UNIVERSITÈ DE STRASBOURG

ÉCOLE DOCTORALE DES SCIENCES DE LA VIE ET DE LA SANTÉ

LABORATOIRE DES SCIENCES DE L'INGÉNIEUR, DE L'INFORMATIQUE ET DE L'IMAGERIE (ICUBE), UMR 7357
INSTITUT DE RECHERCHE MATHÉMATIQUE AVANCÉE (IRMA), UMR 7501

THÈSE présentée par :

Titin Agustin NENGSIH

Soutenue le : 16 mars 2020

pour obtenir le garde de : Docteur de l'Université de Strasbourg

Discipline/Spécialité : Biostatistique

Robustesse et dimensions des modèles de régression PLS en cas de données incomplètes

THÈSE dirigée par :

Pr Nicolas Meyer	**GMRC, Pôle de Santé Publique, CHU de Strasbourg**
Pr Frédéric Bertrand	**Université de Technologie de Troyes**

RAPPORTEURS :

Pr Anne Gégout-Petit	**Université de Lorraine**
Pr Robert Sabatier	**Université de Montpellier**

AUTRES MEMBRES DU JURY :

Pr Erik-André Sauleau	**Université de Strasbourg**
Pr Nicolas Jay	**Université de Lorraine**
Dr Myriam Maumy-Bertrand	**Université de Strasbourg**

Jury

Le jury de cette soutenance de thèse est composé des membres suivants :

- **Pr Nicolas Meyer**, directeur de thèse
- **Pr Anne Gégout-Petit**, rapporteur externe
- **Pr Robert Sabatier**, rapporteur externe
- **Pr Erik-André Sauleau**, examinateur interne
- **Pr Nicolas Jay**, examinateur externe
- **Dr Myriam Maumy-Bertrand**, examinateur externe

Liste des articles et des conférences

1. Article

Determining the Number of Components in PLS Regression on Incomplete Data Set,
T.A. Nengsih, F. Bertrand, M. Maumy-Bertrand et N. Meyer,
Statistical Applications in Genetics and Molecular Biology, 2019, **18**(6),
DOI :10.1515/sagmb-2018-0059.

2. Conférences internationales

Communications orales

- *Determining the Number of Components of a PLS Regression for MAR Mechanism*,
  T.A. Nengsih, F. Bertrand, M. Maumy-Bertrand et N. Meyer,
  The 18th Annual Conference of the European Network for Business and Industrial Statistics,
  du 2 au 6 septembre 2018 à Nancy, France.
- *Influence of Missing Data on Determining the Number of Components for a PLS Regression on MCAR and MAR mechanism*,
  T.A. Nengsih, F. Bertrand, M. Maumy-Bertrand et N. Meyer,
  The 19th Annual Conference of the European Network for Business and Industrial Statistics,
  du 2 au 4 septembre 2019 à Budapest, Bulgarie.

Communications par affiche

- *Influence of Missing Data on the Estimation of the Number of Components of a PLS Regression*,
  T.A. Nengsih, F. Bertrand, M. Maumy-Bertrand et N. Meyer,
  The 17th Applied Stochastic Models and Data Analysis,
  du 6 au 9 juin 2017 à Londres, Angleterre.
- *A Comparison of Determining the Number of Components of a PLS Regression for MCAR mechanism*,

T.A. Nengsih, F. Bertrand, M. Maumy-Bertrand et N. Meyer,
10èmes Assises France-Indonesie : *The Joint Working Group for Cooperation in Higher Education, Research and Innovation*,
du 26 au 28 juin 2018 à Poitiers, France.

- *Determining the Number of Components for a PLS Regression on Incomplete Data*,
  T.A. Nengsih, F. Bertrand, M. Maumy-Bertrand et N. Meyer,
  The 23rd International Conference on Computational Statistics,
  du 28 au 31 août 2018 à Lasi, Roumanie.

- *The Performance of Different Algorithms to Determine the Number of Components for a PLS Regression on MCAR and MAR mechanism*,
  T.A. Nengsih, F. Bertrand, M. Maumy-Bertrand et N. Meyer,
  Statlearn 2019,
  du 4 au 5 avril 2019 à Grenoble, France.

3. Journée École Doctorale

Communications par affiche

- *Determining the Number of Components for a PLS Regression on Incomplete Data for MCAR assumption*,
  T.A. Nengsih, F. Bertrand, M. Maumy-Bertrand et N. Meyer, *Doctoral School Days*,
  du 8 au 9 mars 2018 au Collège Doctoral Européen de l'Université de Strasbourg, France.

4. Rapport de recherche

- *Système de recommandation : algorithmes et application à la plateforme KeeSeek*,
  V. Agniel, F. Bertrand, E. Claeys, A. Delyon, M. Maumy-Bertrand et T.A. Nengsih,
  Semaine Étude Maths-Entreprise,
  du 12 au 16 novembre 2018 à Strasbourg, France.

Remerciements

Il me sera très difficile de remercier ici toutes les personnes que j'aimerai complimenter car elles sont nombreuses mais c'est grâce à l'aide de toutes ces personnes que j'ai pu mener cette thèse à sa fin.

Je souhaite d'abord remercier mon directeur de thèse, le Professeur Nicolas Meyer, d'avoir accepté l'encadrement de cette thèse malgré toutes les difficultés que nous avons rencontrées pendant ces trois années.

Je tiens également à remercier mon co-directeur de thèse, le Professeur Frédéric Bertrand, qui m'a dirigé, encadré pendant ces trois années et ce de façon quotidienne. Je le remercie aussi pour m'avoir fait partager ses brillantes idées. Je souhaite également le remercier pour sa gentillesse, sa patience légendaire, sa disponibilité permanente et pour ses nombreux encouragements qui m'ont permis d'avancer dans ce travail de recherche. Je suis ravie d'avoir travaillé en sa compagnie car outre son appui scientifique, il a toujours été là pour me soutenir moralement et me conseiller au cours de l'élaboration de la rédaction de ce manuscrit. Merci à vous Frédéric !

Je tiens également à remercier le Professeur Anne Gégout-Petit et le Professeur Robert Sabatier pour avoir accepté le rôle de rapporteur de ma thèse, rôle qui n'a pas du être facile car mon français n'est pas toujours parfait. J'adresse également mes remerciements à Erik-André Sauleau et Nicolas Jay en ayant accepté le rôle d'examinateurs et également m'avoir écouté pendant les comités de mi-thèse.

J'exprime toute ma gratitude à Myriam Maumy-Bertrand d'avoir également accepté d'être un de mes examinateurs, et surtout de m'avoir accueilli pendant trois années de suite au sein de l'IRMA. Un énorme merci à Myriam d'avoir réussi à allier la recherche théorique et la recherche appliquée au cours de cette thèse. Cette thèse est aussi le fruit d'une collaboration de trois années avec Myriam. C'est à ses côtés que j'ai enfin compris ce que signifiaient rigueur et précision. C'est Myriam qui m'a aidé à rédiger, à progresser et à me poser les bonnes questions. Myriam, je ne vous remercierai jamais assez !

Merci à toute l'équipe *PMU 5000 Doktor*, Monsieur Mastuki, Madame Salamah Agung et Madame Yeni Ratna Yuningsih, de m'avoir aidé pendant ces années de thèse, notamment sur des questions administratives liées à la bourse. Ce travail n'aurait pas été possible sans le soutien financier du Ministère des affaires religieuses de l'Indonésie, attribué dans le cadre de ce programme *PMU 5000 Doktor* : *MORA Scholarship*. Grâce à une allocation de recherches et diverses aides financières, je me suis consacrée sereinement à l'élaboration de ce manuscrit de thèse. Ce travail de recherche n'aurait également pu être mené à bien sans l'aide et le soutien du

campus UIN Sulthan Thaha Saifuddin Jambi où j'ai travaillé jusqu'à présent. Je tiens également à remercier les présidents d'UIN Sulthan Thaha Saifuddin Jambi : le Professeur Hadri Hasan (2011 - 2019) et le Professeur Su'aidi Asy'ari (2019 - aujourd'hui) pour toute l'aide que vous avez su m'apporter.

Je remercie mon cher époux, Raden Surya Darma, pour son soutien quotidien et son enthousiasme contagieux pendant ses trois années de recherche. Merci de me soutenir chaque jour et de prendre tant soin de moi. Merci de ton amour, de si bien me comprendre et d'être toujours autant attentionné malgré les années.

Mes remerciements vont aussi à mes enfants, Syifa et Rifqi, vos rires et vos blagues ont rendu ma vie plus colorée et c'est vraiment ce soutien dont j'ai besoin chaque jour.

À la mémoire de mon cher Papa, Chairuddin Yahya, c'est à lui que je dois mes beaux jours d'enfance. Papa, tu m'as dévoilé la beauté de la vie et tu m'as donné la joie de vivre.

Je remercie également ma très chère Mère, Yusnimar, qui a toujours été là pour moi et qui m'a toujours encouragé. Vous avez su croire en moi et m'avez apporté toute son aide quand j'en ai eu besoin.

Je remercie mes sœurs, Neneng Yuliarti et Chynta Aprillia, et mes frères, Ahmad Yani et Hardiyan, pour toutes leurs prières, leurs encouragements, leur amour et leur soutien perpétuel. Les mots ne sont pas assez forts pour décrire toute l'aide qu'ils m'ont apporté. Je leur suis infiniment reconnaissante.

Je remercie également l'ensemble du personnel du Département de Mathématique de l'université de Strasbourg et les bibliothécaires pour leur gentillesse. Je pense notamment à Alexis Palaticky, Alain Sartout, Matthieu Boileau et David Brusson pour leur précieuse aide en informatique.

Je voudrais aussi exprimer ma reconnaissance envers mes amis et mes collègues de l'IRMA et d'ICube, notamment Emmanuelle, Marie, Enise, Xing Lu, Viet-Cuong et Marianne, qui m'ont apporté leur soutien moral, intellectuel et tolérant tout au long de ces trois années.

Un grand merci à Anne Westermann, Cécile Gottié, l'équipe ARES et l'équipe SAINT-VINCENT Strasbourg pour leur chaleureuse amitié et leurs conseils concernant mes fautes d'orthographe et de grammaire et également les remercier pour l'opportunité qu'ils m'ont offerte de suivre le cours de français. Cela m'a beaucoup aidé dans mon travail de rédaction. Je remercie également pour les belles sorties autour de Strasbourg afin que je puisse visiter et découvrir les belles villes d'Alsace. Une merveilleuse expérience pour ma famille !

Je tiens aussi à remercier mes amis indonésiens (Uni Neni, Kang Kusna, Kak Dwi, Kak Gianto, Dinda, Ari, Mbak Aisyah, Mas Tesla, Bang Djaffar, Mbak Iin, Putri et Hydra) qui, par leur aide ou leur sympathie ont participé à la réalisation de cette thèse. Et plus je remercie aussi tous les amis de l'association d'étudiants indonésiens de Strasbourg pour leur chaleureuse amitié. Pour mes collègues de *MORA Scholarship* 2016, « Bon Courage à Tous.... » De plus, pour avoir

partagé mon quotidien, mes humeurs...et parfois mon désordre, je remercie mes collègues à l'UIN Sulthan Thaha Saifuddin Jambi, notamment Uni Agustina, Yuk Elyanti, Mas Aris, Mbak Irma et Mas Ayub.

Merci à mes deux familles adorées, la famille Yahya et la famille Raden, qui ne comprennent peut-être pas ce que je fais, mais qui sont très fières ! Merci de votre amour.

Merci aussi à ma belle-famille, à tous mes beaux-frères, mes belles-sœurs et leurs enfants, pour tous leurs encouragements et pour la joie que vous partagez avec moi !

Enfin, Merci Dieu ! ! Ta présence à côté de moi, que je ne mérite pas, est une force. Et je tiens à remercier tous les membres de ma famille et tous mes amis qui ne sont pas mentionnés ici par manque de place, pour leur soutien et pour l'énergie dans nos actions passées, présentes et à venir.

Notations et abréviations

Liste des notations

Notation	Signification
$(.)'$	Transposition
n	Nombre d'unités statistiques qui constituent l'échantillon
p	Nombre de variables indépendantes qui constituent la matrice $\mathbf{X}$
q	Nombre de variables indépendantes qui constituent la matrice $\mathbf{Y}$
$\mathbb{R}^n$	$\mathbb{R}$-espace vectoriel de dimension n
$\mathbf{X}$	Matrice des données $(n \times p)$ pour les variables explicatives
$\mathbf{Y}$	Matrice des données $(n \times q)$ pour les variables réponses
$\mathbf{T}$	Matrice des composantes $(n \times H)$
$\mathbf{W}$	Matrice des coefficients des variables $\mathbf{X}$ dans chaque composante $\mathbf{T}$
$\mathbf{P}$	Matrice des poids $(p \times H)$ de $\mathbf{X}$
$\mathbf{C}$	Matrice de poids $(q \times H)$ de $\mathbf{Y}$
$\mathbf{t}_h$	Vecteur colonne de $\mathbf{T}$ sur la composante h
$\mathbf{w}_h$	Vecteur colonne de $\mathbf{W}$ sur la composante h
$\mathbf{p}_h$	Vecteur colonne de $\mathbf{P}$ sur la composante h
$\mathbf{c}_h$	Vecteur colonne de $\mathbf{C}$ sur la composante h
H	Nombre de composantes retenues
t^*	Nombre vrai de composantes en simulation
d	Proportion de données manquantes
$\mathscr{N}$	Loi normale

Liste des abréviations

Liste des abréviations en anglais

Abréviation	Signification
KNNimpute	*Imputation based on K-Nearest Neighbor*
MAR	*Missing At Random*
MCAR	*Missing Competly At Random*
MICE	*Multiple Imputation by Chained Equations*
MNAR	*Missing Not At Random*
NIPALS	*Nonlinear Iterative PArtial Least Squares*
PLS	*Partial Least Squares*
SVDimpute	*Imputation based on Singular Value Decomposition*

Liste des abréviations en français

Abréviation	Signification
ACP	Analyse en Composantes Principales
ACPP	Analyse en Composantes Principales Probabiliste
DM	Données Manquantes
MCO	Moindres Carrées Ordinaires
NIPALS-PLS	*PLS* utilisant l'algorithme *NIPALS*
RCP	Régression sur Composantes Principales

Table des matières

Liste des figures

Liste des tableaux

Première partie

La problématique

Chapitre 1

Mise en place de la problématique

1.1 Le contexte

Actuellement personne ne peut nier que les développements et les innovations technologiques dans tous les domaines scientifiques émergent tous les jours et aux quatre coins du monde. En effet, les recherches publiques ou privées génèrent de plus en plus d'expériences complexes et produisent de plus en plus de données. D'ailleurs, quotidiennement, les médias nous rappellent que nous vivons dans un déluge de données. Nous pouvons citer pour illustrer nos propos, à titre d'exemple les domaines où les données s'accumulent en masse (cette liste ne se veut pas exhaustive) : l'agriculture, l'astronomie, la biologie, la chimie, la médecine, la météorologie, la physique, les sciences de l'ingénieur et les sciences sociales.

Par exemple, lors d'une étude prospective, le chercheur suivra le protocole expérimental avec rigueur et minutie afin de récolter de nouveaux résultats qui reflètent le plus possible la réalité. Mais cela reste de l'expérimentation et pour pouvoir généraliser les premières conclusions obtenues, la maîtrise de la modélisation statistique est un élément indispensable pour établir une théorie que le chercheur tente de mettre en place.

Dans cette thèse, nous nous sommes focalisés sur la recherche et l'innovation dans les sciences du vivant. En effet, il est indéniable que depuis ces deux dernières décennies, les méthodes en biostatistique se sont développées rapidement, et ce, grâce aux nouvelles expériences mises en place par les biologistes et/ou les médecins. La collaboration entre ces disciplines (la biologie, la médecine, les mathématiques et même l'informatique) est très encouragée. D'ailleurs, à titre d'exemple, rappelons les nombreux appels à projets émanant de ces disciplines. De ces collaborations naissent de nouvelles méthodes et avancées et nous pouvons citer à titre d'exemple les articles suivants : en allélotypage (Meyer *et al.*, 2010), en génomique (Vallat *et al.*, 2013; Bastien *et al.*, 2015; Liquet *et al.*, 2016), en métabolique (Liew *et al.*, 2011) et en multi-omique (Fornecker *et al.*, 2019).

Nous sommes donc amenés à nous poser la question : pourquoi en sommes-nous là ? Un premier élément de réponse est que la biologie a fait des avancées extraordinaires en enrichissant sa connaissance et en s'appuyant sur des analyses statistiques très pertinentes. En effet, les statisticiens qui collaborent avec le monde du vivant ont introduit des modèles beaucoup plus

sophistiqués qui s'adaptent mieux aux résultats biologiques. De plus, la masse de données a conduit les statisticiens à développer de nouveaux modèles pour décrire et analyser ces données. En parallèle, l'accroissement de la puissance de calcul des ordinateurs a également permis l'implémentation de certains modèles théoriques pour répondre à des phénomènes biologiques complexes. Depuis ces dernières années, les logiciels de statistiques comme *SAS*, *Stata* et *SPSS* ont introduit de nouveaux algorithmes pour résoudre les problèmes théoriques pour lesquels nous n'avions pas de solutions numériques rapides. Le langage R s'est également développé et de nouveaux `packages` de R permettent de répondre à des problématiques spécifiques liées aux sciences du vivant. Citons à titre d'exemple le `package` `plsRglm` implémenté dans le langage R et développé par Bertrand, Meyer et Maumy-Bertrand en 2014 (Bertrand *et al.*, 2014).

1.2 Pourquoi la régression linéaire ne convient-elle pas dans notre problématique ?

De nombreuses expériences émanant du monde du vivant peuvent être modélisées par un modèle de régression. En effet, le biologiste dispose de variables explicatives (stockées dans une matrice notée $\mathbf{X}$) sur lesquelles il peut plus ou moins agir et a identifié une(des) variable(s) réponse(s) (stockée(s) également dans une colonne ou dans une matrice notée $\mathbf{Y}$). L'objectif est donc de décrire et de modéliser les relations existantes entre la (les) variable(s) réponse(s) et les variables explicatives. Pour résoudre ce type de problème, le biologiste peut par exemple utiliser dans un premier temps un modèle linéaire.

Nous avons à notre disposition un échantillon sur lequel nous avons observé des variables. Nous supposons ici que toutes ces **variables** sont **quantitatives** et que la **réponse** est **univariée**. Dans la suite, nous noterons n le nombre d'observations, $\mathbf{y}$ le vecteur réponse de dimension $n \times 1$ et $\mathbf{X}$ la matrice de taille $n \times p$ (les lignes de la matrice correspondent aux n observations et les colonnes aux p variables explicatives).

Les observations d'un individu ou d'une unité statistique i sont réunies sous la forme d'un $(p+1)$-uplet noté $(x_{i1}, x_{i2}, \ldots, x_{ip}, y_i)$ pour i allant de 1 à n. Pour simplifier les notations, nous écrirons $(\mathbf{x}_{i\ldots}, y_i)$ pour désigner le $(p+1)$-uplet ci-dessus. $\mathbf{x}_{i\ldots}$ désigne donc la $i^{\text{ème}}$ ligne de la matrice $\mathbf{X}$. Les observations de la variable $\mathbf{x}_j$ sont données par le vecteur colonne $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})'$ qui correspond à la colonne j de la matrice $\mathbf{X}$.

Par définition, le modèle linéaire spécifie la relation linéaire qui peut exister entre la variable réponse et l'ensemble des variables explicatives. L'objectif est donc de construire ce modèle (par conséquent d'estimer les paramètres de ce modèle) qui permet d'expliquer la variation de la réponse à partir des p variables explicatives. Parfois ce modèle peut être utilisé pour prédire la valeur possible de la réponse à partir de nouvelles observations. La plupart du temps, le modèle linéaire s'écrit :

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \tag{1.1}$$

où $\beta = (\beta_0, \ldots, \beta_p)'$ est le vecteur composé des $(p+1)$ coefficients du modèle linéaire et ε le terme d'erreur.
L'estimation des coefficients du modèle linéaire utilise la méthode des moindres carrés ordinaires.

Cette dernière correspond à une minimisation du terme d'erreur ε, i.e. (Hastie *et al.*, 2009) :

$$\hat{\beta} = \min_{\beta \in \mathbb{R}^{p+1}} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 \tag{1.2}$$

où $\|.\|_2$ désigne la norme ℓ_2. En résolvant le problème de minimisation ci-dessus lorsque le modèle est régulier, c'est-à-dire lorsque la matrice $\mathbf{X}$ est de rang maximal, l'estimateur du vecteur β est égal à :

$$\hat{\beta} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y}$$

où $(\mathbf{X}'\mathbf{X})^{-1}$ représente l'inverse de la matrice $\mathbf{X}'\mathbf{X}$.

Si les variables explicatives sont non corrélées linéairement entre elles, et ce, de façon significative, si elles sont peu nombreuses et si elles ne présentent pas de données manquantes, alors l'estimation par les moindres carrés ordinaires du modèle linéaire se révèle être la méthode la plus adaptée pour ajuster un modèle linéaire à ce type de données. Ainsi, nous avons identifié trois conditions fréquentes d'application du modèle linéaire. Cependant si l'une de ces trois conditions n'est pas vérifiée, alors le modèle linéaire se révèle inadapté.

1.3 Les problèmes liés au jeu de données

1.3.1 La multicolinéarité entre les variables

Souvent, dans la pratique, un certain nombre de variables explicatives peuvent être très fortement corrélées linéairement, nous parlons alors de multicolinéarité. Il existe plusieurs indicateurs de multicolinéarité. La multicolinéarité est la principale source d'instabilité dans l'estimation des coefficients du vecteur β. La présence de ces forts liens linéaires entre les variables explicatives implique directement que les valeurs propres de la matrice $\mathbf{X}'\mathbf{X}$ sont alors proches de zéro. Par conséquent, l'estimation des coefficients de la régression linéaire multiple par la méthode des moindres carrés ordinaires n'est plus envisageable.

Une autre conséquence de la présence de multicolinéarité est que l'inverse de la matrice aura des valeurs très grandes. Nous aurons alors recours à une matrice inverse généralisée mais l'utilisation de cette dernière entraîne la perte de l'unicité de l'estimateur $\hat{\beta}$. En d'autres termes, l'estimation individuelle de tous les coefficients du vecteur de paramètres β n'est plus possible. Comme le rappelle Tenenhaus dans Tenenhaus (1998), tout cela aura pour conséquence une incohérence au niveau des signes des estimations des coefficients du modèle et parfois l'absence de significativité de certaines variables explicatives alors qu'elle serait pourtant attendue.

Il existe plusieurs indicateurs de la multicolinéarité. L'approche la plus classique consiste à examiner les facteurs d'inflation de la variance, abrégés en *FIV* (*variance inflation factor* en anglais). Les *FIV* estiment l'augmentation ou la diminution de la variance d'un estimateur du coefficient de la variable explicative en fonction de la relation linéaire avec les autres variables explicatives. En règle générale, Si les *FIV* sont supérieurs à 10 alors les variables explicatives sont très corrélées linéairement entre elles (H. Kutner *et al.*, 2004) (un seuil de 5 est également couramment utilisé (J. Sheather, 2009)). Si les *FIV* sont plus proches de 1, alors le modèle est

beaucoup plus robuste, car les variables ne sont pas influencées par les corrélations avec d'autres variables.

1.3.2 Les dimensions du jeu de données

En bioinformatique, le traitement des jeux de données de grande dimension (ici nous rappelons qu'un jeu de données est un tableau comportant p colonnes et n lignes) comme c'est le cas, par exemple, avec les puces à ADN, les *microarrays* ou les données de séquençage à haut débit, reste un problème.

Nous pouvons également citer le domaine de la chimiométrie. Les chimiomètres doivent traiter des jeux de données où il y a plus de variables (de 1000 à 5000) que d'observations (de 10 à 100). Le nombre d'observations est souvent limité pour des raisons différentes, comme par exemples des raisons éthiques, des raisons financières, etc. Cette limite sur les observations est de plus en plus fréquente.

Dès que le nombre de variables dépasse strictement le nombre d'observations ($p > n$), alors la matrice $\mathbf{X}'\mathbf{X}$ n'est pas inversible. La matrice carrée de variance-covariance est donc singulière. La présence d'un très grand nombre de variables explicatives peut amener ainsi une certaine complexité dans l'interprétation des résultats.

1.3.3 Les valeurs manquantes

Le fait que les valeurs manquantes (absence de résultats de l'expérience) sont de plus en plus présentes dans la recherche scientifique est incontournable. Les données manquantes sont fréquentes dans le processus de collecte des observations. En effet, certaines contraintes de terrain provoquent cette situation comme la main d'œuvre quasiment inexistante pour récolter les données, l'impossibilité d'envoyer ou de récupérer les questionnaires, l'incompréhension du questionnaire, les erreurs d'équipement, de mesure, le manque de financement pour acheter du matériel ou encore l'impossibilité d'acheter des animaux de laboratoire, etc.

Le traitement inadéquat de ces données manquantes a plusieurs conséquences. En analyse statistique, il peut conduire par exemple à des estimations biaisées comme celles des paramètres d'un modèle et en particulier les coefficients d'un modèle de régression linéaire multiple. En parallèle, cela entraîne également une estimation biaisée des écarts-types de ces estimateurs de paramètres et ainsi une mauvaise construction d'intervalles de confiance pour les coefficients du modèle. C'est pour cette raison qu'il existe de nombreuses études méthodologiques qui étudient les propriétés des méthodes d'imputation pour les données manquantes dans les recherches biologiques, et ce, dans différentes conditions (White *et al.*, 2011; Bouhlila and Sellaouti, 2013; Schmitt *et al.*, 2015; Pedersen *et al.*, 2017). Étant donné le nombre grandissant de méthodes d'imputation, il nous parait important de consacrer le prochain chapitre de ce manuscrit à des rappels sur les méthodes d'imputation (le tout sera expliqué au chapitre 2).

1.4 Quelques solutions pour résoudre ces trois problèmes

1.4.1 Comment résoudre les problèmes de multicolinéarité et de dimension ?

Pour résoudre les problèmes de multicolinéarité entre les variables et de dimension du jeu de données, plusieurs solutions sont possibles. Nous avons choisi volontairement de citer les deux plus utilisées, à savoir la régression sur composantes principales, et la régression des moindres carrés partiels ou régression par projection sur des structures latentes, abrégée en régression *PLS* (*Partial Least Square* en anglais). Ces deux méthodes de régression sont très populaires dans certains domaines, notamment en chimie dans le développement de la spectroscopie (Goicoechea and Olivieri, 2003). Une des raisons principales de la popularité de ces méthodes est qu'elles ont été spécialement développées pour traiter les problèmes issus de ces domaines qui ont des expériences avec de nombreuses variables majoritairement très corrélées entre elles et très peu d'observations.

La régression sur composantes principales

La régression sur composantes principales, introduite par Hotelling (1957), permet la décomposition de la matrice représentant le jeu de données, notée $\mathbf{X}$, contenant les p variables corrélées linéairement. Cette méthode va construire de nouvelles composantes principales qui sont en réalité des combinaisons linéaires des variables initiales et qui, par construction ne sont pas corrélées linéairement entre elles. C'est le même principe que dans une analyse en composantes principales. Ainsi nous pourrons ensuite effectuer une régression multiple entre la(les) variable(s) réponse(s) et les composantes principales construites à partir de la matrice $\mathbf{X}$. Un autre objectif est d'assurer une variance maximale afin de représenter au mieux les variables explicatives de la matrice $\mathbf{X}$.

Pour atteindre ces deux objectifs, nous rappelons rapidement comment nous procédons : nous construisons les H (où $H < \text{rg}\mathbf{X}$) composantes notées $\mathbf{t}_H$ en diagonalisant la matrice de variance-covariance $\mathbf{X}'\mathbf{X}$. Ainsi nous récupérons les H vecteurs propres notés $(\mathbf{w}_1, \ldots, \mathbf{w}_H)$ qui sont associés aux valeurs propres notées $(\lambda_1 \geqslant \ldots \geqslant \lambda_H)$ ordonnées de la plus grande à la plus petite valeur. Ainsi nous avons, pour tout $h = 1, \ldots, H$:

$$\mathbf{t}_h = \mathbf{X}\mathbf{w}_h. \tag{1.3}$$

Rappelons que l'analyse en composantes principales cherche un ensemble de vecteurs noté $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_H)$ qui maximisent la variance des nouvelles composantes principales, rappelons-les, notées $(\mathbf{t}_1, \ldots, \mathbf{t}_H)$. Nous avons donc pour la composante h :

$$\mathbb{V}\text{ar}(\mathbf{t}_h) = \frac{1}{n-1}\mathbf{t}_h{}'\mathbf{t}_h = \frac{1}{n-1}\mathbf{w}_h{}'\mathbf{X}'\mathbf{X}\mathbf{w}_h, \tag{1.4}$$

en ajoutant le problème d'optimisation suivant :

$$\mathbf{w}_h = \underset{w \in \mathbb{R}^p}{\operatorname{argmax}} \, \mathbf{w}_h{}' \mathbf{X}' \mathbf{X} \mathbf{w}_h \tag{1.5}$$

sous la contrainte

$$\mathbf{w}' \mathbf{w} = 1.$$

Ensuite nous continuons le processus en effectuant une régression linéaire multiple entre la réponse $\mathbf{y}$ et les composantes principales issues de cette analyse en composantes principales. Le problème majeur engendré par la régression sur composantes principales est que les premières composantes, associées aux plus grandes valeurs propres, ne sont pas nécessairement corrélées avec la réponse $\mathbf{y}$. Elles ne sont donc pas nécessairement les meilleures candidates pour expliquer ou prédire $\mathbf{y}$ (Jolliffe, 1982).

D'ailleurs, nous le constatons bien dans le processus : ces composantes principales ne prennent pas en compte dans leur construction de la réponse $\mathbf{y}$. Elles maximisent uniquement la qualité de représentation de la matrice $\mathbf{X}$. Ainsi l'inconvénient de cette méthode est que la première composante principale n'est pas forcement celle qui expliquera le mieux la réponse $\mathbf{y}$. Bien qu'il existe la procédure de choix des composantes les plus corrélées à la variable explicative par exemple régression sur composantes principales pénalisées (Byrd, 2008). Il faudrait plutôt choisir les composantes principales les plus corrélées linéairement avec la réponse $\mathbf{y}$ (Lee *et al.*, 2015). Ce qui implique de changer de point de vue et de méthode !

La régression des moindres carrés partiels

La régression des moindres carrés partiels, abrégée en régression *PLS*, est une généralisation de la régression linéaire multiple. Contrairement à cette dernière, la régression *PLS* permet d'analyser des variables explicatives très corrélées linéairement sans commettre des erreurs de signes des coefficients ou de significativité pour les coefficients. Elle peut également modéliser simultanément un ensemble de variables réponses. En résumé, la régression *PLS* est considérée comme un compromis entre la régression par les moindres carrés ordinaires et la régression sur composantes principales.

La régression *PLS* est une méthode dans laquelle les paramètres du modèle sont estimés à l'aide de l'algorithme *Straightforward Implementation of a statistically inspired Modification to PLS*, abrégé en *SIMPLS*, développée par De Jong (De Jong, 1993b), ou de l'algorithme *Nonlinear Iterative PArtial Least Squares*, abrégé en *NIPALS* (Wold, 1966). Ces algorithmes créent des variables latentes (composantes non corrélées) qui sont construites comme des combinaisons linéaires des variables explicatives issues de la matrice $\mathbf{X}$. Par ailleurs, ces composantes maximisent la covariance avec la variable réponse $\mathbf{y}$, ce que ne faisait pas la régression sur composantes principales.

Pour un nombre fixé de composantes, De Jong (1993a) a montré que l'estimation $\mathbf{y}$ produite par la régression *PLS* est liée à une erreur d'approximation plus faible que la régression sur composantes principales. La comparaison des performances prédictives a été faite par Frank and Friedman (1993) et Krämer and Sugiyama (2011). Ces derniers montrent que la régression *PLS*

a besoin de moins de composantes que la régression sur composantes principales pour atteindre des performances similaires.

Pour conclure sur ces méthodes présentées brièvement, la régression *PLS* a un grand avantage à savoir elle construit des composantes maximisant la covariance avec la réponse $\mathbf{y}$ tout en gardant les propriétés d'une régression. Pour finir la régression linéaire multiple maximise la corrélation, la régression sur composantes maximise la variance et la régression *PLS* maximise la covariance.

1.4.2 Comment résoudre les valeurs manquantes ?

Rappelons qu'une façon de traiter les manquants (aujourd'hui certaines personnes continuent de la faire) consiste à éliminer la(les) variable(s) qui présente(nt) des valeurs manquantes et la(les) observation(s) qui n'est(sont) pas relevées. Cette méthode de suppression est largement critiquée actuellement car elle génère une perte énorme d'informations. Rappelons que dans certains domaines, la taille du jeu de données est déjà petite donc avec cette technique, la taille de l'échantillon devient inappropriée pour certaines méthodes statistiques. En effet, réduire la taille de l'échantillon conduit à une réduction importante en termes de puissance statistique. Il faut donc encore une fois changer de méthode et se tourner vers une autre : l'imputation.

L'utilisation de méthodes statistiques et de calcul numérique va permettre d'imputer les valeurs manquantes. En effet, de nombreuses études ont montré que la fusion d'informations provenant de diverses sources peut améliorer l'estimation des valeurs manquantes. L'objectif de l'imputation est donc d'utiliser des relations déjà existantes ou qui peuvent être identifiées dans les observations non manquantes pour remplacer celles qui manquent. Il est à noter qu'il existe plusieurs méthodes d'imputation. Nous citerons ici quelques méthodes :

1. l'imputation par la moyenne (Troyanskaya *et al.*, 2001),
2. l'imputation par la médiane. Ces deux méthodes sont applicables quand les données sont quantitatives,
3. l'imputation basée sur la régression linéaire multiple (Horton and Lipsitz, 2001),
4. l'impution multiple (Rubin, 1987),
5. la méthode des plus proches voisins (en anglais *K-nearest neighbor*, *KNN*) qui est notée *KNNimpute* (Troyanskaya *et al.*, 2001),
6. la méthode d'imputation par décomposition en valeurs singulières (en anglais *singular value decomposition*, *SVD*) qui notée *SVDimpute* (Troyanskaya *et al.*, 2001), etc.

Une autre façon de gérer les données manquantes est d'utiliser une méthode qui gère les valeurs manquantes. En 1966, Wold (Wold, 1966) a introduit l'algorithme *NIPALS*. Ce dernier a largement été utilisé en génomique en raison de son efficacité pour modéliser des jeux de données où les variables sont corrélées et/ou les dimensions du jeu sont grandes et notamment le cas où le nombre d'unités statistiques est inférieur au nombre de variables ($n < p$). Il a l'avantage de pouvoir construire les composantes même lorsque les données sont incomplètes. En effet,

chaque composante est construite à partir des seules données complètes, de manière itérative sur chaque dimension du jeu de données et ceci, sans devoir recourir à l'imputation.

Nous présenterons ces méthodes de traitement des données manquantes brièvement dans le chapitre 2 dans la Section 2.3.

D'après la brève explication donnée ci-dessus, nous pouvons dire que la régression *PLS* s'appuyant sur l'algorithme *NIPALS* est une méthode qui semble résoudre les trois problèmes exposés précédemment. Le fait d'avoir associé la régression *PLS* et l'algorithme *NIPALS* a donné, ce qui est appelé, la régression *NIPALS-PLS*. Elle se révèle donc comme une méthode d'analyse des données et de modélisation qui permet de traiter le cas particulier des jeux de grandes dimensions, avec des variables explicatives éventuellement fortement corrélées et potentiellement incomplètes. De plus, Andersson (2009) a comparé neuf algorithmes issus de la méthode *PLS* dans le cas de la réponse univariée. Il a montré que l'algorithme stable est, semble-t-il, l'algorithme *NIPALS* de Wold. Il est donc très intéressant d'approfondir la régression *PLS* utilisant l'algorithme *NIPALS* en particulier sur le cas de données incomplètes. Le chapitre 3 présentera brièvement les généralités de la régression *PLS* sur des données complètes et sur des données incomplètes.

1.5 Objectifs de la thèse

Bien que l'algorithme *NIPALS* soit désormais considéré comme une méthode de référence dans le traitement des données manquantes, les performances de cet algorithme ont été très peu étudiées dans le cas de données incomplètes jusqu'à aujourd'hui. De plus, la sensibilité de l'algorithme *NIPALS* par rapport à la proportion de données manquantes ne semble pas avoir été étudiée jusqu'à maintenant.

La détermination du nombre de composantes construites lors de la régression *NIPALS-PLS* ne tient pas compte ni du mécanisme de manquant ni de la proportion de données manquantes dans le jeu de données. Pourtant il s'agit d'un point essentiel pour établir des modèles de régression robustes ainsi que pour sélectionner correctement des variables explicatives.

La détermination du nombre de composantes d'un modèle de régression *PLS* sur données incomplètes est le sujet principal de cette thèse. La détermination du nombre de composantes est choisie en fonction des critères utilisés pour construire un modèle de régression *PLS*. Plusieurs critères ont été étudiés, notamment les critères de la validation croisée et les critères d'information avec les simulations. Le chapitre 4 présentera un certain nombre de critères de sélection de modèles de régression *NIPALS-PLS*. Dans le chapitre 5, nous présenterons le plan de nos simulations.

Ainsi, l'objectif de cette thèse consiste à déterminer le nombre de composantes d'une régression *PLS* sur des données incomplètes qui sont liées à certaines caractéristiques des données que nous avons considérées comme importantes lors des simulations. Ces caractéristiques sont :

- les dimensions du jeu de données $n > p$ (matrice verticale) et $n < p$ (matrice horizontale),
- le nombre vrai de composantes,

- le mécanisme de données manquantes,
- la proportions de données manquantes,
- les critères.

Pour ce faire, nous avons étudié d'abord les performances globales de la régression *NIPALS-PLS* pour déterminer le nombre de composantes sans données manquantes (les données complètes). Nous présenterons ces résultats dans le chapitre 6. Ensuite, nous examinerons l'effet du mécanisme et de la proportion de données manquantes sur la régression *NIPALS-PLS*.

Il nous est apparu intéressant de comparer les performances des critères sur un jeu de données incomplet et sur un jeu de données imputé, ou plus simplement effectuer une comparaison entre la régression *NIPALS-PLS* sur des données incomplètes et la régression *NIPALS-PLS* sur de données imputées. Les trois méthodes d'imputation que nous avons étudiées sont :

- l'imputation multiple Rubin (1987) par l'imputation *Multivariate Imputation By Chained Equations*,
- l'imputation des plus proches voisins (Troyanskaya *et al.*, 2001),
- l'imputation par décomposition en valeurs singulières (Troyanskaya *et al.*, 2001).

Nous présenterons les résultats des comparaisons de ces méthodes dans le chapitre 7. Enfin, nous avons testé ces trois méthodes d'imputation sur des données réelles provenant de différentes expériences. Le pré-traitement de données réelles présentera dans le chapitre 8. Cela nous a permis ainsi d'examiner en détails les propriétés que nous avions observées dans le cas de données simulées. Nous présenterons ces résultats dans le chapitre 9.

Deuxième partie

La méthodologie statistique

Chapitre 2

Méthodes statistiques pour le traitement des valeurs manquantes

Il faut noter que parfois, dans certaines conditions d'expérience, des valeurs ou des mesures ne sont pas observées ou relevées. Ce phénomène de manquant est alors présent dans beaucoup de domaines où l'analyse des données est un outil incontournable. Les méthodes statistiques permettant de traiter les données manquantes sont de plus en plus nombreuses car elles se développent rapidement. Pour plus de détails sur ce sujet, nous renvoyons le lecteur aux références suivantes : Little and Rubin (1987), Little and Rubin (2002), Schafer (1997), Schafer and Graham (2002), Royston (2004), Dempster *et al.* (1997), Hastie *et al.* (1999), Mandel J (2015), etc. Toutes ces méthodes sont alors développées et présentes dans les logiciels de statistique. Nous pouvons d'ailleurs citer à titre d'exemple le logiciel libre  qui permet gérer les données manquantes en proposant plusieurs *packages*.

Définir ce qu'est une donnée et une donnée manquante est assez complexe et converger vers une seule définition est un vrai défi. La définition peut même dépendre parfois du domaine dans lequel les données sont relevées. Les données manquantes peuvent apparaître ou pas en fonction de l'expérience, des conditions de l'expérience, du mécanisme de l'expérience dont nous disposons. La première section de ce chapitre (voir Section 2.1) va tenter de définir le terme «données» et l'expression «données manquantes», pour être ensuite utilisés dans la suite de mon manuscrit de thèse. La deuxième section de ce chapitre (voir Section 2.2) pose le cadre théorique du mécanisme sous-jacent de l'apparition de manquants. La troisième section de ce chapitre (voir Section 2.3) répertorie un certain nombre de méthodes utilisées pour gérer les données manquantes. Cette section ne vise pas à fournir toutes les méthodes de traitement des données manquantes. En effet, nous nous sommes intéressés uniquement aux solutions qui correspondent à notre cadre de recherche. À la fin de ce chapitre (voir Section 2.4), nous présenterons quelques *packages* du logiciel libre  et ces derniers traiteront les données manquantes dans les cadres de recherche qui nous ont intéressées.

2.1 Définitions

Nous allons tenter de définir «données » et «données manquantes». D'ailleurs à ce sujet, nous renvoyons à la thèse de Sciences médicales de Nicolas Meyer et en particulier aux pages 53 et 54 (Meyer, 2007).

DEFINITION 2.1.1 *Les* **données** *sont une série de mesures ou de relevés sur une ou plusieurs unité(s) statistique(s) provenant d'une (ou de) variable(s) à laquelle (auxquelles) nous nous intéressons. Ces mesures peuvent être de type quantitatif ou qualitatif.*

REMARQUE 2.1.1 *Cette collection d'informations est la plupart du temps mise dans un tableau à deux dimensions, noté* $\boldsymbol{X}$*. Ce tableau* $\boldsymbol{X}$ *peut donc être assimilé à une matrice et contient toutes les valeurs possibles de la (des) variable(s) étudiée(s).*

DEFINITION 2.1.2 *Les* **données** *sont dites* **complètes** *lorsque le tableau qui les contient a toutes ses cases remplies avec la valeur observée de la (des) variable(s) étudiée(s).*

DEFINITION 2.1.3 *Les* **données** *ou* **valeurs** *sont dites* **manquantes** *lorsque le tableau qui les contient a un certain nombre de cases vides, c'est-à-dire la valeur de l'observation de la (des) variable(s) n'a pas été relevée (éventuellement pour différentes raisons).*

DEFINITION 2.1.4 *Les* **données** *ou* **valeurs** *sont dites* **imputées** *lorsque la valeur manquante a été remplacée par un processus d'imputation.*

REMARQUE 2.1.2 *Il existe plusieurs méthodes d'imputation pour obtenir des valeurs imputées. Il peut donc avoir plusieurs choix possibles pour cette valeur à imputer.*

Maintenant que nous avons défini la notion de «données manquantes», nous allons nous intéresser au mécanisme statistique de celles-ci.

2.2 Mécanisme statistique menant aux valeurs manquantes

Rappelons la définition de «donnée manquante» énoncée par Nicolas Meyer dans Meyer (2007) : nous appelons donnée manquante une donnée pour laquelle nous ne disposons pas de la valeur de l'observation de la variable étudiée pour une unité statistique donnée. Le problème de la gestion des données manquantes est alors un sujet lui-même. Selon les domaines et les compétences statistiques du praticien, il y a plusieurs solutions qui s'offrent au praticien (nous ne nous voulons pas exhaustifs sur la liste ci-dessous) :

1. supprimer les variables (les colonnes de la matrice $\mathbf{X}$) et/ou les observations (les lignes de la matrice $\mathbf{X}$) qui présentent des données manquantes, ou
2. imputer des valeurs aux données manquantes, ou
3. utiliser ou développer des algorithmes robustes qui permettent de mener les analyses statistiques en présence de valeurs manquantes.

Soit $\mathbf{X} \in \mathbb{R}^{n\times p}$ une matrice rectangulaire contenant les données. L'entier p représente le nombre de variables explicatives, notées $\mathbf{x}_1, \ldots, \mathbf{x}_p$ et l'entier n le nombre d'observations.
Introduisons $\mathbf{R} = (r_{ij})$ la matrice indiquant la présence ou l'absence des valeurs manquantes dans la matrice $\mathbf{X}$.
Nous notons les valeurs observées par : $\mathbf{X}_{obs}$ et si l'élément x_{ij} de la matrice $\mathbf{X}$ est observé alors l'élément r_{ij} de la matrice $\mathbf{R}$ est égal à 1.
Nous notons les valeurs manquantes par : $\mathbf{X}_{manq}$. Par conséquent l'élément r_{ij} de la matrice $\mathbf{R}$ est égal à 0.
Ainsi, la matrice $\mathbf{X}$ se décompose ainsi : $(\mathbf{X}_{obs}, \mathbf{X}_{manq})$.
Avec ces notations, le mécanisme des données manquantes est caractérisé par la distribution conditionnelle de la matrice $\mathbf{R}$ sachant $\mathbf{X}$, notée $(\mathbf{R}|\mathbf{X})$.

Nous allons maintenant présenter les trois types de mécanisme statistique des valeurs manquantes, définis par Rubin en 1976 dans Rubin (1976) puis présentés dans Little and Rubin (1987) :

- Les **valeurs** sont dites **manquantes de façon complètement aléatoire** (en anglais *Missing Completely At Random*), abrégé en *MCAR*, si la probabilité d'une valeur manquante ne dépend ni des mesures observées ni des non observées. Cela se traduit mathématiquement par :

  $$\mathbb{P}(\mathbf{R} = 0|\mathbf{X}) = \mathbb{P}(\mathbf{R} = 0).$$

  Ceci implique que la présence de valeurs manquantes n'est pas reliée au processus d'observations. Par conséquent la probabilité d'être manquant est alors la même pour tous les cas. Enfin, il est à noter, d'après Besse (2020), que le mécanisme *MCAR* est peu courant.

  Nous allons donner ci-dessous quelques exemples.

  EXEMPLES 2.2.1

  - *un relevé de température où les valeurs manquantes correspondent à une panne du thermomètre,*
  - *une mesure de pression artérielle pour laquelle les valeurs manquantes correspondent à un tensiomètre cassé.*

- Les **valeurs** sont dites **manquantes de façon aléatoire** (en anglais *Missing At Random*), abrégé en *MAR*, si la probabilité d'une valeur manquante ne dépend que des mesures observées. Cela se traduit mathématiquement par :

  $$\mathbb{P}(\mathbf{R} = 0|\mathbf{X}) = \mathbb{P}(\mathbf{R} = 0|\mathbf{X}_{obs}).$$

  Nous remarquons que dans ce cas nous utilisons uniquement l'information observée. De plus, le processus *MAR* est beaucoup plus fréquent que le processus *MCAR*. En effet, il est plus général et plus réaliste que le processus *MCAR*.

  EXEMPLES 2.2.2

  - *Prenons l'exemple de la réalisation d'une enquête sur la dépression nerveuse chez l'adulte (hommes et femmes). Nous pouvons presque affirmer que les manquants suivent un mécanisme MAR, et ce parce que le genre des patients influe les réponses. En effet, les hommes sont perçus comme moins sensibles que les femmes et donc*

moins sujets à la dépression. Quant à elles, les femmes remplissent plus facilement le questionnaire. Elles veulent se soigner car elles sont souvent mères et ne se permettent pas d'être une malade chronique. Une fois que le genre est pris en compte dans l'analyse des données, la disparité ne dépend pas du niveau de dépression.

- *Prenons un autre exemple : nous souhaitons étudier le poids chez les adultes. Nous savons que les relevés de poids chez les jeunes seront moins remplis que chez les sujets plus âgés, parce que les jeunes adultes fréquentent moins souvent les établissements de santé que les adultes plus âgés.*

- Les **valeurs** sont dites **manquantes de façon non aléatoire** (en anglais *Missing Not At Random*), abrégé en *MNAR*, si la probabilité d'une valeur manquante dépend de la variable en question. Cela se traduit mathématiquement par :

$$\mathbb{P}(\mathbf{R} = 0|\mathbf{X}) \quad \text{ne se simplifie pas.}$$

Il faut donc noter que la probabilité d'être manquant dépend aussi de l'information non observée. Le processus *MNAR* est le cas le plus complexe. Les stratégies pour traiter ce type de manquants sont de trouver le maximum d'informations et d'explications sur les causes des manquants.

EXEMPLES 2.2.3

- *Les personnes ayant un revenu très élevé refusent en général de répondre aux questions portant sur leur revenu.*
- *Une réponse proposée parmi tant d'autres à une question d'un questionnaire est souvent évitée car le statut engendré par cette réponse est difficile à assumer par la personne interrogée.*

Dans cette thèse, nous nous sommes intéressés qu'aux deux premiers mécanismes puisque le troisième nécessite de spécifier le mécanisme matériel (non statistique) d'apparition des valeurs manquantes, ce qui requiert un traitement adapté à chacune situation.

2.3 Méthodes de traitement des valeurs manquantes

Plusieurs approches statistiques ont été développées pour traiter les valeurs manquantes de type *MCAR* et *MAR*. Nous allons présenter ci-dessous les méthodes de traitement des valeurs manquantes qui peuvent être classées en quatre catégories.

2.3.1 Méthode avec suppression des valeurs

L'analyse des cas complets (en anglais *Complete Case Analysis*) est une méthode largement utilisée pour traiter les données manquantes. Cette méthode, également appelée « *list-wise deletion* », consiste à exclure tout sous-ensemble incomplet du tableau de données ou éliminer toutes les observations ayant des données manquantes pour les variables étudiées.

EXEMPLE 2.3.1 *Si un individu présente une seule valeur manquante pour une seule variable, alors la ligne représentant l'individu sera entièrement supprimée du tableau de données.*

Cette méthode est largement utilisée pour traiter toutes les situations car elle est facile à implémenter. C'est souvent l'option par défaut dans la plupart des logiciels de statistique comme *SPSS*, *SAS* et *Stata*. Dans R, il faut utiliser la fonction `na.omit()`.

REMARQUE 2.3.1 *Lorsque les données sont de type MCAR, cette méthode produira des estimations non biaisées pour la moyenne, la variance et les poids dans une régression pondérée (Van Buuren, 2012). Cependant, cette méthode peut diminuer la qualité du modèle résultant si le nombre d'observations complètes n'est pas assez élevé.*

L'inconvénient majeur de cette méthode est le gaspillage des données en termes de temps et d'argent... En utilisant cette méthode, nous avons souvent constaté que plus de la moitié des observations sont perdues, surtout si le nombre p de variables est grand. Il est évident que le sous-échantillon construit à partir de la suppression d'observations peut sérieusement réduire la capacité de détecter l'effet cherché puisqu'il n'est peut-être plus représentatif de l'effet cherché.

Si les données ne sont pas de type *MCAR*, l'analyse des cas complets peut fortement biaiser l'estimateur de la moyenne, l'estimateur du coefficient de corrélation linéaire. Little and Rubin (2002) montrent que le biais de l'estimateur de la moyenne augmente avec la différence entre la moyenne des observations et des valeurs manquantes et avec la proportion des données manquantes. Schafer and Graham (2002) ont également étudié cette méthode en faisant des simulations sous les mécanismes *MAR* et *MNAR*.

Ainsi, le cas de type *MAR*, ignorer de données manquantes introduit un biais dans la modélisation. En effet, dans ce cas la valeur d'une variable manque lorsque d'autres variables, observées, prennent certaines combinaisons d'autres valeurs. Ignorer les observations incomplètes revient à ignorer la plupart des occurrences de ces combinaisons de valeurs pour ces variables observées. Il faut donc plutôt d'imputer les observations incomplètes par l'estimation de l'imputation des données manquantes grâce à une modélisation de la dépendance entre les variables observées et la variable à valeurs manquantes.

Dans le cas de type *MNAR*, la suppression des observations à données manquantes introduit également un biais dans la modélisation car cela revient à éliminer des observations de façon non aléatoire. De plus l'absence de la valeur d'une variable ne dépend pas de variables observées, l'imputation des données manquantes à partir des valeurs prises par les variables observées est donc difficile à justifier.

2.3.2 Méthode d'estimation des paramètres

Une méthode d'estimation des données manquantes consiste à effectuer des itérations successives en ré-estimant de la même manière les modèles statistiques qui sont liés aux relations entre les variables. Les données manquantes étant elles-mêmes ré-estimées en maximisant les critères associés aux modèles statistiques et aux données observées. Ce cadre algorithmique a été développé par Dempster *et al.* (1997) en mettant en œuvre cette idée. Cet algorithme s'appelle l'algorithme *expectation–maximization*, abrégé en *EM* pour la suite.

L'algorithme *EM* est une approche générale du calcul itératif de vraisemblances optimales en présence de données manquantes. Des vraisemblances sont calculées de manière itératives en mettant à jour l'estimation des données manquantes. D'après Van Buuren (2012), l'approche basée sur la vraisemblance définit un modèle à partir des données observées. Ainsi, il n'y a pas besoin d'imputer les données manquantes ou d'écarter les observations incomplètes. Les inférences sont basées sur la vraisemblance selon le modèle posé. Les paramètres sont estimés par la méthode maximum de vraisemblance par l'algorithme *EM*.

Cet algorithme se divise en deux étapes qui sont :

- **Étape** *E*, étape d'estimation, les données manquantes sont estimées en utilisant une espérance conditionnelle sur les données présentes et les paramètres estimés à l'itération précédente.
- **Étape** *M*, étape de maximisation, Les paramètres sont calculés en maximisant la vraisemblance d'utiliser les données estimées à l'**étape** *E* en plus des données existantes.

Ainsi, les méthodes basées sur la vraisemblance sont dans un sens la voie royale pour traiter les données incomplètes. L'utilisation du maximum de vraisemblance consiste à maximiser la probabilité de l'estimateur sur toutes les données complètes et incomplètes. Les paramètres estimés résument bien l'information disponible sous les modèles présumés pour les données complètes et les données manquantes également. Les hypothèses du modèle peuvent être affichées et évaluées, et dans la plupart des cas il est possible d'estimer l'erreur standard des estimations.

Des descriptions formelles pour le développement de cette méthode sont fournies dans le Chapitre 10. Cette partie ne présente pas comme le travail principal mais comme un support de réflexion au développement des objectifs de la recherche à venir.

2.3.3 Méthodes d'imputation

L'imputation est une technique de remplacement des données manquantes par une valeur générée lors du processus d'imputation. L'objectif est d'utiliser des relations connues ou qui peuvent être identifiées dans les valeurs présentes des jeux de données pour aider à estimer les données manquantes. Nous présentons ci-après quelques méthodes d'imputation qui sont fréquemment utilisées par les praticiens.

Imputation simple. L'imputation simple consiste à remplacer chaque donnée manquante par une estimation de sa valeur et à analyser la base de données ainsi complétée. L'imputation simple est la solution simple et rapide. Cependant, par exemple l'imputation par le moyenne conduit à une sous-estimation parfois de la variance des estimateurs et biaisera l'estimation de la moyenne lorsque les données ne sont pas des *MCAR* (Van Buuren, 2012).

Les valeurs utilisées pour cette imputation sont :

1. Une valeur fixe. C'est-à-dire que nous remplaçons par la même valeur toutes les variables à données manquantes, par exemple la valeur zéro (0). Cette solution est un choix arbitraire

qui doit être évitée. L'utilisation de la valeur 0 est concevable si la variable concernée est de moyenne nulle.

2. Une valeur représentative de la distribution de la variable concernée :

   - Pour une variable quantitative, les données manquantes sont remplacées par une seule valeur, comme la moyenne. Ce processus s'appelle l'imputation par la moyenne. La moyenne des valeurs présentes (non manquantes) est toutefois sensible à la présence de quelques valeurs extrêmes, le choix de la médiane est donc préférable.

     Dans le cas d'imputation par la moyenne pour une variable : soit $x_1, x_2, \ldots, x_a$ sont les données observées et $x_{a+1}, x_{a+2}, \ldots, x_n$ sont les données manquantes, chaque valeur manquante est remplacée par $\bar{x}_{obs}$.

     $$\bar{x}_{obs} = \frac{1}{a}\sum_{i=1}^{a} x_i. \tag{2.1}$$

     L'imputation par la moyenne devrait être utilisée uniquement lorsque quelques valeurs sont manquantes et devrait en général être évitée (Van Buuren, 2012).

   - Un autre choix est l'imputation par la médiane. La médiane des valeurs non manquantes a une bien meilleure robustesse que la moyenne en présence de valeurs extrêmes.

   - Les variables quantitatives peuvent prendre la valeur de Mode pour remplacer les données manquantes. C'est-à dire qu'ils ne peuvent prendre qu'un nombre très limité de valeurs différentes ou pour une variable nominale (à modalités). Il est possible de choisir la valeur la plus fréquente parmi les valeurs présentes.

Imputation par la régression linéaire. Cette méthode consiste à remplacer la ou les donnée(s) manquante(s) en réalisant la régression de la variable qui présente la donnée à imputer sur un certain nombre de variables sans données manquantes. La première étape consiste à construire un modèle à partir des données observées. Les prévisions pour les cas incomplets sont ensuite calculées à partir du modèle approprié et ces prévisions remplaceront alors la ou les donnée(s) manquante(s).

La condition fondamentale de l'imputation par régression linéaire est ainsi l'existence d'une relation linéaire entre les variables. En pratique, un modèle est produit à partir des données observées, puis les données manquantes sont remplacées par les valeurs de prédictions de ce modèle. Cette méthode est alors utilisée de manières itérées et chaque valeur d'imputation est réutilisée pour être réalisée avec d'autres valeurs.

L'imputation par la régression linéaire donne des estimations non biaisées de la moyenne sous l'hypothèse *MCAR*. Les poids de régression ne sont pas biaisés sous l'hypothèse *MAR*, si les facteurs qui influencent les données manquantes font partie du modèle de régression (Little and Rubin, 2002). De plus, l'imputation des valeurs prédites par régression peut donner des imputations réalistes si la prédiction est proche de la perfection (Van Buuren, 2012). En fait, une relation linéaire entre les variables est une hypothèse rarement vérifiée.

Imputation multiple. Depuis ces dernières années, l'imputation multiple est reconnue comme la meilleure méthode pour traiter les données manquantes. Cette méthode a été proposée pour la première fois par Rubin (1987), puis développée par Little and Rubin (1987) et reprise par Schafer (1997). L'utilisation de l'imputation multiple est également importante en raison de sa généralité et des récents développements des moyens de calcul informatique. Son utilisation est en augmentation constante.

Le concept d'imputation multiple repose sur l'utilisation de la distribution de données observées pour estimer un ensemble de valeurs plausibles pour remplacer les données manquantes. Cela revient à résoudre un problème avec des données manquantes, en analysant séparément plusieurs problèmes avec des données complètes (Van Buuren, 2012).

L'imputation multiple crée $m \geqslant 1$ jeux de données complètes. Rubin (1987) a publié un article qui recommande le nombre de $m = 5$ imputations. Chacun de ces jeux de données est alors analysé par des méthodes classiques, puis une combinaison de leurs résultats est réalisée pour tenir compte aussi bien des données initialement observées que des données imputées ainsi que de l'incertitude associée à cette imputation.

En utilisant les recommandations de Rubin (1987), les m résultats de l'analyse statistique sont combinés en une unique estimation et donnent une erreur standard. Les trois étapes de l'imputions multiple sont décrites ci-dessous : l'imputation, l'analyse statistique et la combinaison de ces m estimations. La Figure 2.1 reprend ces trois étapes.

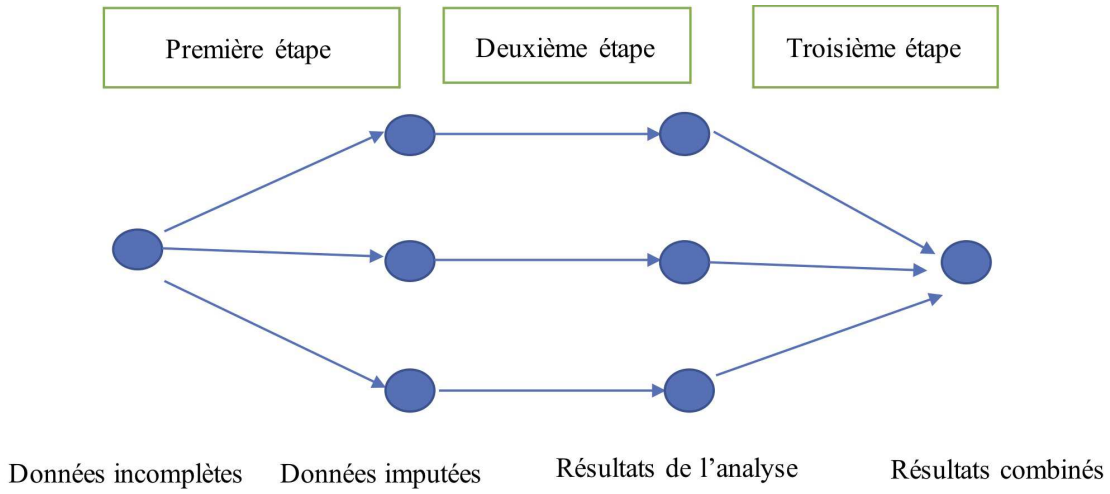


Figure 2.1 – Les trois étapes de l'imputation multiple avec $m = 3$.

1. Première étape : l'imputation.
   L'analyse commence par de données observées qui sont incomplètes. L'étape d'imputation commence à créer m groupes de ce jeu de données en substituant les données manquantes par des données plausibles. Ces valeurs plausibles sont tirées d'une distribution spécialement modélisée pour chaque donnée manquante. La figure 2.1 présente $m = 3$, ce qui signifie qu'il y a 3 jeux de données imputés. En pratique, m est souvent choisis plus grand. Le nombre $m = 3$ est pris simplement pour indiquer que la technique crée plusieurs versions

des données imputées. Les trois jeux de données imputés sont différents car l'ampleur de cette différence reflète l'incertitude quant à la valeur à imputer.

2. Deuxième étape : l'analyse statistique sur chaque jeu de données imputées.
   La deuxième étape consiste à estimer les paramètres d'intérêt de chacun des m jeux de données imputées. Sur chaque jeu de données imputées, nous appliquons la méthode statistique standard. Les m résultats obtenus seront tous différents puisque les données imputées sont également différentes.

3. Troisième étape : la combinaison.
   La dernière étape consiste à combiner les m estimations du paramètre en une seule estimation, et d'estimer sa variance. La variance combine la variance intra-imputation et la variance entre-imputations qui est une variance supplémentaire provoquée par les données manquantes. Lorsque les conditions appropriées, les estimations regroupées sont non biaisées et possèdent les propriétés statistiques correctes (Van Buuren, 2012).

Des détails du sujet de l'imputation multiple sont disponibles notamment dans les travaux de Rubin (1987), Schafer (1997) et Van Buuren (2012). Au cours de cette thèse, nous n'avons pas étudié spécifiquement ce problème. Cependant, il semble important que nous utilisions des processus capables de gérer ces données manquantes.

Le choix du nombre m de groupes n'est pas arrêté. En effet, les choix usuels pour m sont $m = 3$, $m = 5$ et $m = 10$. Certains auteurs ont même étudié l'effet de m sur divers aspects des résultats. Il est souvent recommandé d'augmenter m, à savoir le mettre dans une plage de 20 à 100 imputations. De plus amples détails sur le choix de m sont fournis, entre autres, par Schafer (1997), Royston (2004), Graham *et al.* (2007), Bodner (2008) et White *et al.* (2011).

Selon White *et al.* (2011), il y a une règle très simple que le nombre d'imputation m devrait être similaire au pourcentage de données manquantes. De plus, Van Buuren (2012) affirme qu'il est préférable d'utiliser un m élevé, même si cela implique plus de calculs et de stockage et cela peut être utile pour les estimations qui rapprochent de la distribution complète.

L'imputation multiple par *MICE* (Van Buuren and Groothuis-Oudshoorn, 2011) est une approche pratique pour créer des imputations. Elle est souvent utilisée dans la première étape de l'imputation multiple. L'algorithme *MICE* commence avec un tirage aléatoire à partir des données observées, et impute les données incomplètes variable par variable.

La méthode *MICE* est aussi connue comme une méthode de spécification conditionnelle totale (en l'anglais *Fully Conditional Specification*) (Van Buuren, 2007). L'algorithme *MICE* a été développé et a été étudié plus en détails par Van Buuren and Groothuis-Oudshoorn (2000) et Van Buuren and Groothuis-Oudshoorn (2011). Initialement, toutes les données manquantes sont remplacées par un échantillonnage aléatoire simple avec remise des données observées. Dans cette thèse, le nombre m de jeux de données imputées sera égal à $100 \times$ la proportion de valeurs manquantes, comme (White *et al.*, 2011) le recommande.

Imputation *KNN*. L'approche de l'imputation *KNN*, abrégée en *KNNimpute*, est une méthode d'imputation qui est basée sur les plus proches voisins par l'algorithme *KNN*. Cet algorithme a

pour but de regrouper les observations en fonction de leur similarité (Batista and Monard, 2003). Il nécessite le choix du paramètre K par optimisation d'un critère, par exemple la distance.

Pour les données manquantes, considérer que les observations complètes les plus proches sont plus représentatives que les autres constitue une hypothèse moins restrictive. La proximité doit être déterminée à partir de calculs de distance limités aux seules variables renseignées pour l'observation de données manquantes. Ainsi, les données manquantes sont imputées en utilisant des valeurs qui sont calculées à partir des K voisins les plus proches. La valeur imputée est une valeur, comme la moyenne, qui correspond aux K voisins les plus proches. C'est une méthode très simple et puissante.

En général, la procédure pour chaque observation de données manquantes est :

1. trouver ses K plus proches voisins (observations complètes) en tenant compte, dans les calculs de distances, uniquement des valeurs des variables renseignées pour cette observation,
2. donner comme valeur, à chaque variable non renseignée, la moyenne des valeurs que prend la même variable pour ces K voisins.

Le choix du paramètre K se fait par optimisation d'un critère. Troyanskaya *et al.* (2001) ont montré que la valeur K optimale se situe entre 10 et 20. Nous avons pris K égal à 15.

Il y a plusieurs types de distances qui peuvent être utilisés pour mesurer les similitudes entre deux individus, tels que la corrélation de Pearson, la distance de Mahalanobis, la distance de Chebyshev, la distance euclidienne et la distance de Gower. Hastie *et al.* (1999) et Troyanskaya *et al.* (2001) ont utilisé une distance basée sur la similarité euclidienne.

Imputation SVD**.** La méthode SVD est une méthode avec une réduction de rang qui fournit une bonne approximation des observations (complètes). C'est-à-dire, lorsque les observations décrites par p variables quantitatives se situent à proximité d'un sous-espace linéaire de dimension j bien plus faible que p. Dans ce cas, une estimation de la donnée manquante d'une observation est obtenue à l'intersection entre le sous-espace qui est une bonne approximation des observations complètes et le sous-espace obtenu en fixant les valeurs des variables connues pour l'observation avec des données manquantes. L'imputation SVD, abrégé en $SVDimpute$ est étudiée et mis en œuvre dans le contexte de données de *microarray* à ADN également par Troyanskaya *et al.* (2001).

En général, $SVDimpute$ peut être appliquée sur des données manquantes de la manière suivante : tout d'abord, nous initialisons toutes les données manquantes par colonne (variable) avec la moyenne de cette variable ou la valeur zéro (0). Ensuite, jusqu'à la convergence, nous effectuons l'imputation SVD de la matrice complète. Enfin, nous remplaçons les données manquantes par leurs estimations à partir des imputation SVD.

Une décomposition en valeurs singulières, de rang réduit j, est appliquée à la matrice complète, notée $\mathbf{X}_{comp}$ (Hastie *et al.*, 1999) par :

$$\hat{\mathbf{X}}_{(comp)j} = \mathbf{U}_j \mathbf{D}_j \mathbf{V}'_j, \tag{2.2}$$

où $\mathbf{U}_j$ est une matrice $n \times j$ qui a pour colonnes les vecteurs propres de $\mathbf{X}_{comp}(\mathbf{X}_{comp})'$ correspondant aux j plus grandes valeurs propres. $\mathbf{D}_J$ est une matrice diagonale $j \times j$ comprenant les j valeurs propres de $\mathbf{X}_{comp}(\mathbf{X}_{comp})'$ en ordre décroissant. $\mathbf{V}_j$ est une matrice $p \times j$ qui a pour colonnes les vecteurs propres de $\mathbf{X}'_{comp}(\mathbf{X}_{comp})$ correspondant aux j plus grandes valeurs propres.

L'imputation des données manquantes est faite par une régression similaire à celle des moindres carrés qui s'écrit :

$$\min_{\beta \in \mathbb{R}^J} \sum_{i \text{ observées}} \left(\mathbf{X}_{i^*} - \sum_{j=1}^{J} v_j \beta_j \right)^2. \tag{2.3}$$

Soit $\mathbf{V}_j^*$ la matrice réduite obtenue à partir de $\mathbf{V}_j$, c'est-à-dire que pour laquelle les lignes correspondant aux données manquantes de la ligne $\mathbf{X}_{i^*}$ sont éliminées. Une solution du problème d'équation (2.3) est alors :

$$\hat{\beta} = \left(\mathbf{V}_j^{*\prime}\mathbf{V}_j^*\right)^{-1}\mathbf{V}_j^{*\prime}\mathbf{X}_{i^*}. \tag{2.4}$$

La prédiction des données manquantes est donc donnée par :

$$\mathbf{X}_{i^*} = \mathbf{V}_j^{(*)}\hat{\beta}, \tag{2.5}$$

où $\mathbf{V}_j^{(*)}$ est le complément de $\mathbf{V}_j^*$ dans $\mathbf{V}_j$. Notons que nous avons pris $j = 10$ dans le choix du paramètre j.

2.3.4 Méthode tolérante vis à vis des données manquantes : l'algorithme *NIPALS*

L'algorithme *NIPALS* permet d'estimer les données manquantes sans l'imputation préalable. Cet algorithme a été proposé par Wold en 1966 (Wold, 1966) et repris par Tenenhaus en 1998 (Tenenhaus, 1998). Le principe général de l'algorithme *NIPALS* consiste à effectuer une séquence itérative de régressions simples permettant de récupérer des données. Ces résultats de régressions simples sont combinés les uns aux autres pour déterminer les principales composantes qui expliquent la diminution de l'information contenues dans les données.

L'algorithme *NIPALS*, algorithme itératif, travaille à partir de la matrice $\mathbf{X}$ qui contient les p variables explicatives linéairement corrélées entre elles (au passage le nombre p peut être supérieur au nombre n d'observations dans le cadre de *NIPALS*). De plus, il faut noter que le rang de la matrice $\mathbf{X}$ est égal à H. Remarquons au passage que $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_p$ représentent les colonnes de la matrice $\mathbf{X}$. De plus, nous supposerons que ces variables seront centrées. La formule de décomposition de $\mathbf{X}$ par l'analyse en composantes principales s'écrit, d'après Tenenhaus (Tenenhaus, 1998) :

$$\mathbf{X} = \sum_{h=1}^{H} \mathbf{t}_h \, \mathbf{p}_h'. \tag{2.6}$$

où $\mathbf{t}_h = (t_{h1}, \ldots, t_{hn})$ sont les vecteurs de composantes principales et $\mathbf{p}_h = (p_{h1}, \ldots, p_{hp})$ sont les vecteurs directeurs des axes principaux.

Les variables $\mathbf{x}_j$ s'expriment en fonction de composantes $\mathbf{t}_1, \ldots, \mathbf{t}_H$:

$$\mathbf{x}_j = \sum_{h=1}^{H} p_{hj}\mathbf{t}'_h, j = 1, \ldots, p. \tag{2.7}$$

La i-ème ligne de $\mathbf{X}$ est notée $\mathbf{x}'_i = (x_{1i}, \ldots, x_{ji}, \ldots, x_{pi})$. Alors les individus $\mathbf{x}_i$ peuvent aussi s'exprimer en fonction des vecteurs $\mathbf{p}_1, \ldots, \mathbf{p}_h$:

$$\mathbf{x}_i = \sum_{h=1}^{H} t_{hi}\mathbf{p}'_h, i = 1, \ldots, n. \tag{2.8}$$

Lorsqu'il n'y a pas de donnée manquante, l'algorithme *NIPALS* conduit à l'analyse en composantes principales (ACP) usuelle. Cependant s'il y a des données manquantes, nous obtenons des valeurs des estimations des composantes $\mathbf{t}_h$ et des vecteurs $\mathbf{p}_h$ qui permettent de décrire la matrice de données $\mathbf{X}$ et d'estimer les données manquantes.

Soit une matrice $\mathbf{X}$ de données complètes, c'est-à-dire qu'il n'y a pas de données manquantes, les étapes de l'algorithme *NIPALS* sont décrites ci-dessous (Tenenhaus, 1998) :

1. $\mathbf{X}_0 = \mathbf{X}$
2. Pour $h = 1, 2, ..., H$:
   (a) $\mathbf{t}_h$ = première colonne de $\mathbf{X}_{h-1}$
   (b) Répéter jusqu'à convergence de $\mathbf{p}_h$
   (c) $\mathbf{p}_h = \mathbf{X}'_{h-1}\mathbf{t}_h / \mathbf{t}'_h\mathbf{t}_h$
   (d) Normer $\mathbf{p}_h$ à 1
   (e) $\mathbf{t}_h = \mathbf{X}_{h-1}\mathbf{p}_h / \mathbf{p}'_h\mathbf{p}_h$
3. $\mathbf{X}_h = \mathbf{X}_{h-1} - \mathbf{t}_h\mathbf{p}'_h$

Chaque coordonnée p_{hj} du vecteur $\mathbf{p}_h$ représente, avant normalisation, le coefficient de régression de $\mathbf{t}_h$ dans la régression de la variable $\mathbf{x}_{h-1,j}$ sur la composante $\mathbf{t}_h$. De plus, chaque coordonnée t_{hi} d'un vecteur $\mathbf{t}_h$ correspond au coefficient de régression de $\mathbf{p}_h$ dans la régression sans constante de $x_{(h-1),i}$ sur $\mathbf{p}_h$.

Pour h = 1, nous obtenons le vecteur propre $\mathbf{p}_1$ et la première composante principale $\mathbf{t}_1$. La matrice $\mathbf{X}_1 = \mathbf{X} - \mathbf{t}_1\mathbf{p}'_1$ représente le résidu de la régression $\mathbf{X}$ sur la première composante principale $\mathbf{t}_1$. Par conséquent, pour h = 2, le vecteur propre $\mathbf{p}_2$ de la matrice $\frac{1}{n-1}\mathbf{X}'_1\mathbf{X}_1$ associé à la plus grande valeur propre correspond au vecteur propre de $\frac{1}{n-1}\mathbf{X}'_1\mathbf{X}_1$ associé à la deuxième plus grande valeur propre λ_2. Ainsi, de manière générale, la matrice $\frac{1}{n-1}\mathbf{X}'_{h-1}\mathbf{X}_{h-1}$ et le vecteur propre $\mathbf{p}_h$ de $\frac{1}{n-1}\mathbf{X}'_{h-1}\mathbf{X}_{h-1}$ sont associés à la h-ième plus grande valeur propre λ_h.

L'intérêt de l'algorithme *NIPALS* apparaît plus clairement en présence de données manquantes. Les étapes de l'algorithme *NIPALS* sur un jeu de données incomplètes peuvent se voir dans l'algorithme ci-dessous.

1. $\mathbf{X}_0 = \mathbf{X}$
2. Pour $h = 1, ..., H$:
   (a) $\mathbf{t}_h$ = première colonne de $\mathbf{X}_{h-1}$
   (b) Répéter jusqu'à convergence de $\mathbf{p}_h$
      i. Pour $j = 1, ..., p$:

$$p_{hj} = \frac{\sum_{i \in I} x_{h-1,ji} t_{hi}}{\sum_{i \in I} t_{hi}^2}, \quad avec \quad I = \left\{ i : x_{ji} \quad et \quad t_{hi} \quad existent \right\}$$

      ii. Normer $\mathbf{p}_h$ à 1
      iii. Pour $i = 1, ..., n$:

$$t_{hi} = \frac{\sum_{j \in J} x_{h-1,ji} p_{hj}}{\sum_{j \in J} p_{hj}^2}, \quad avec \quad J = \left\{ j : x_{ji} \quad existe \right\}$$

   (c) $\mathbf{X}_h = \mathbf{X}_{h-1} - \mathbf{t}_h \mathbf{p}_h^t$

L'idée de cet algorithme est de calculer à chaque fois les pentes des droites de moindres carrés, passant par l'origine, des nuages de points sur les données disponibles (Tenenhaus, 1998). Le problème de l'algorithme est de fournir des pseudo-valeurs propres λ_h définis par la variance $\frac{1}{n-1} \mathbf{t}_h' \mathbf{t}_h$ de la composante $\mathbf{t}_h$.

L'algorithme *NIPALS* permet également d'estimer les données manquantes sans l'imputation préalable. Cette méthode permet ainsi de donner une estimation des données manquantes en utilisant la formule de reconstruction à l'ordre h :

$$\hat{x}_{ij} = \sum_{l=1}^{h} t_{li} p_{lj}. \tag{2.9}$$

2.4 L'imputation en pratique : les *packages* du logiciel R

Le logiciel R peut être téléchargé gratuitement sur le site et son installation se fait avec quelques *packages* de base (The R Core Team, 2014). Puis ensuite nous pouvons choisir d'importer d'autres *packages* en fonction des analyses statistiques que nous réalisons. Nous allons utiliser les trois méthodes d'imputations suivantes : *MICE*, *KNNimpute* et *SVDimpute*. Naturellement ces méthodes sont développées dans des *packages* de R : `mice` (Van Buuren, 2018), `VIM` (Templ *et al.*, 2017), et `bcv` (Perry, 2015) respectivement.

Package MICE
En général, certaines fonctionnalités du *package* `MICE` sont :

- inspecter le modèle de données manquantes,

- imputer les données manquantes *m* fois, résultant en *m* ensembles de données imputées (de données complètes),
- analyser chaque ensemble de données imputées,
- mettre en commun les résultats des analyses répétées.

Ce *package* contient quelques fonctions :

- la fonction `mice`() qui fournit une imputation des données manquantes *m* fois.
- La fonction `with`() qui permet d'analyser les ensembles de données imputées.
- La fonction `pool`() qui permet de combiner les estimations de paramètres.
- La fonction`complete`() qui permet d'exporter des données imputées.
- La fonction `ampute`() qui fournit la génération des données manquantes.

Package `VIM`
Le *package* `VIM` est basée sur une variation de la distance de Gower (Gower, 1971). Ce *package* contient la fonction `kNN`() qui effectue l'imputation des données manquantes avec K voisins les plus proches utilisés. Il permet aussi d'exporter des données imputées.

Package `bcv`
Le *package* `bcv` contient la fonction `impute.svd`() qui fournit une imputation des données manquantes. Cette fonction attribue alors les valeurs manquantes à l'aide d'une approximation de la méthode SVD.

Chapitre 3

Généralités sur la régression *PLS*

3.1 Introduction

En général, la méthode *PLS* a de nombreuses tâches statistiques, telles que les capacités de régression et de classification, mais peut également être utilisé en même temps comme outil descriptif. En outre, la méthode *PLS* peut réduire les dimensions et éliminer la colinéarité entre plusieurs variables explicatives comme l'ACP. Ainsi, elle s'est avérée être une technique très puissante dans de nombreux domaines d'application multidisciplinaire.

En effet, initialement, la méthode *PLS* a été développée à partir de méthodes liées au domaine de l'économétrie. Ensuite, elle est utilisée dans quelques domaines comme en bioinformatique (Nguyen and Rocke, 2004), en sciences sociales (Sawatsky *et al.*, 2015), en spectroscopie moléculaire et biomoléculaire (Oleszko *et al.*, 2017) ou en médecine (Yang *et al.*, 2017), etc. Par conséquent, le développement rapide de cette méthode dans de nombreux domaines a fait que presque tous les logiciels de programmation proposent la méthode *PLS* de leurs outils tels que les logiciels R, *SPSS*, *Matlab*, etc.

Dans la discussion des méthodes liée à la famille *PLS*, nous trouvons plusieurs définitions comme suit (Meyer, 2007).

- **La méthode *PLS*.**
  La méthode *PLS* est équivalente à l'ACP.
- **La régression *PLS*1.**
  La régression *PLS*1 est une régression linéaire d'une variable réponse $\mathbf{y}$ sur plusieurs variables explicatives contenues dans la matrice $\mathbf{X}$ utilisant l'algorithme *PLS*. Dans la suite de cette thèse, nous désignerons par régression *PLS* la régression *PLS*1 si il n'y a pas d'ambiguïté.
- **La régression *PLS*2.**
  La régression *PLS*2 est une régression linéaire d'au moins deux variables réponse $\mathbf{y}$ sur plusieurs variables explicatives contenues dans la matrice $\mathbf{X}$ utilisant l'algorithme *PLS*.
- **L'analyse discriminante *PLS*.**

L'analyse discriminante *PLS* est une application de la méthode *PLS* pour l'analyse discriminante.

- **L'analyse canonique *PLS*.**
  L'analyse canonique *PLS* est une application de la méthode *PLS* dans la version canonique. Elle met en relation les variables **Y** et les variables **X**.

3.2 Historique de régression *PLS*

La régression *PLS* est apparue dans les sciences sociales, et plus précisément dans l'économétrie avec Herman Wold qui a développé la méthode *PLS* en 1966 (Wold, 1966). Cette méthode consiste en une suite itérative de régressions linéaires simples en utilisant les moindres carrés ordinaires pour construire les composantes principales (Wold, 1973). L'algorithme de la méthode *PLS* est généralement basé sur l'algorithme *NIPALS* et est apparu en 1977 en tant que procédure répétitive pour rechercher des variables latentes ou des composantes. Une méthode d'estimation alternative est l'algorithme *SIMPLS* de Jong (De Jong, 1993b).

Svante Wold, fils de Herman Wold et chimiste, a donné son point de vue sur le développement de la régression *PLS* dans les années 80 (Wold *et al.* (1983) ; Wold *et al.* (1984) ; Wold *et al.* (1989)). De plus, Stone and Brooks (1990) ont publié la régression avec des prédictions construites à partir de séquences de validation croisée sur la régression linéaire multiple, la régression *PLS* et la régression sur composantes principales.

Quelques années plus tard, la régression *PLS* continue d'être publiée et devient une procédure habituelle en Chimiométrie tel qu'en atteste la publication de Wold *et al.* (2001). Ces travaux ont ensuite transformé la méthode *PLS* en un outil d'analyse de données généraliste dans le domaine de la chimie et peut également être utilisée avec des jeux de données de grandes dimensions (Wold, 2001). Il est à noter que la littérature autour de de la méthode *PLS* abonde, et ce, dans divers domaines scientifiques. De plus, un livre en français entièrement consacré à cette méthode a été écrit par Tenenhaus en 1998 (Tenenhaus, 1998).

3.3 Régression *PLS*

La régression *PLS* est la nouvelle méthode de régression qu'il faut utiliser. Contrairement à la régression linéaire multiple, elle peut analyser des données avec des variables explicatives **X** qui sont très corrélées linéairement, et également de modéliser simultanément plusieurs variables de réponse, **Y**. En l'absence du vecteur/matrice **Y**, la régression *PLS* se réduit alors à une analyse en composantes principales. La régression *PLS* peut aussi être vue comme un compromis entre la régression par les moindres carrées ordinaires et la régression sur composantes principales.

Les avantages de régression *PLS* est qu'elle permet de traiter un jeu de données dans lequel le nombre de variables est supérieur au nombre d'observations du jeu de données. À titre d'exemple, citons le cas de données génomiques ou de données métaboliques. D'ailleurs, elle apparaît comme une méthode multivariée qui peut supprimer de la colinéarité de données. Elle

donne de meilleurs résultats que la régression linéaire multiple lorsque les variables explicatives sont fortement corrélées linéairement (Wold *et al.*, 1984).

Un autre avantage est l'existence de l'algorithme *NIPALS* dans la régression *PLS*. L'algorithme *NIPALS* permet d'obtenir des estimations sans imputer les données manquantes (Tenenhaus, 1998). Cet algorithme a été conçu pour effectuer une ACP sur un jeu de données incomplètes. La régression *NIPALS-PLS* fournit même des estimations des paramètres du modèle de régression multiple dans le cas de données manquantes.

L'objectif principal de la régression *PLS* est de construire un modèle linéaire comme le modèle de la régression linéaire multiple (équation (1.1)). En général, les variables contenues dans la matrice $\mathbf{X}$ sont centrées (nous soustrayons la moyenne), et réduites (nous les divisons par leur écart-type respectif). Nous appliquerons le même processus au vecteur $\mathbf{y}$.

La régression *PLS* produit des composantes (comme dans l'Analyse en Composantes Principales) qui sont des combinaisons linéaires des variables explicatives provenant de la matrice $\mathbf{X}$, de telle manière qu'il n'y a pas de corrélation entre les composantes construites par le modèle de régression. Le nombre de ces composantes peut être inférieur au nombre de variables explicatives contenues dans la matrice $\mathbf{X}$.

Le modèle de régression *PLS*1 se décompose de la façon suivante :

$$\mathbf{y} = \mathbf{T}\mathbf{c} + \mathbf{f} \tag{3.1}$$

avec

$$\mathbf{T} = \mathbf{X}\mathbf{W}^*, \tag{3.2}$$

où $\mathbf{c}$ est un vecteur des coefficients de la régression de $\mathbf{y}$ sur $\mathbf{T}$ et $\mathbf{f}$ est un vecteur des résidus. $\mathbf{T}$ est une matrice des composantes où les colonnes de $\mathbf{T}$, notées $\mathbf{t}_h$, sont indépendantes (au sens orthogonales). $\mathbf{W}$ est une matrice de poids. Nous reviendrons un peu plus tard sur l'obtention des poids $\mathbf{W}^*$.

Le modèle de l'équation (3.1) peut donc s'écrire de la façon suivante :

$$\mathbf{y} = \mathbf{X}\mathbf{W}^*\mathbf{c} + \mathbf{f} \tag{3.3}$$

Le modèle de la régression *PLS*1 peut être illustré à la Figure 3.1, par exemple nous avons pris h égal à trois.

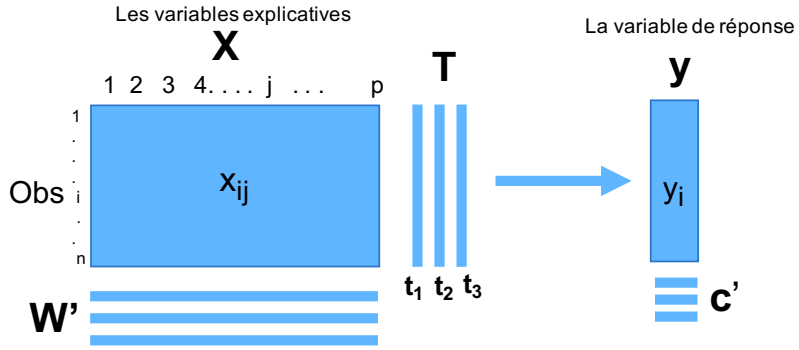


Figure 3.1 – Illustration de la régression *PLS*, par exemple $h = 3$.

3.3.1 Régression *PLS*1 en données complètes

Nous commençons par montrer le principe de l'algorithme de la régression *PLS*1 en présence de données complètes selon l'algorithme *NIPALS*.

Principe de l'algorithme de la régression *PLS*1 et passage à l'écriture matricielle

Le principe général de l'algorithme de la régression *PLS*1 cherche à réaliser la régression d'une variable réponse $\mathbf{y}$ sur p variables explicatives collectées dans une matrice $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$. Ces variables peuvent être fortement corrélées linéairement entre elles.

La régression *PLS*1 se distingue par ses composantes, qui maximisent la covariance entre les variables contenues dans la matrice $\mathbf{X}$ et le vecteur $\mathbf{y}$ (Höskuldsson, 1988). Ces composantes peuvent s'écrire ainsi :

$$\mathbf{t}_h = \mathbf{X}\mathbf{w}_h^*, \tag{3.4}$$

où $h = 1, \ldots, H$.

Algorithme de la régression *NIPALS-PLS*

Nous considérons le modèle de régression *PLS*1 défini dans l'équation (3.1). Pour plus de détails sur chaque composante construite, veuillez vous référer comme suit.

Notons $\mathbf{y}$ et $\mathbf{X}$ comme les valeurs de la variable centrées - réduites. Nous construisons tout d'abord la première composante

$$\mathbf{t}_1 = w_{11}\mathbf{x}_1 + \cdots + w_{1p}\mathbf{x}_p, \tag{3.5}$$

où

$$w_{1j} = \frac{cov(\mathbf{x}_j, \mathbf{y})}{\sqrt{\sum_{j=1}^{p} cov^2(\mathbf{x}_j, \mathbf{y})}}. \tag{3.6}$$

Puis, nous effectuons une régression simple de $\mathbf{y}$ sur $\mathbf{t}_1$ par :

$$\mathbf{y} = c_1\mathbf{t}_1 + \mathbf{y}_1, \tag{3.7}$$

où c_1 est le coefficient de régression et $\mathbf{y}_1$ est le vecteur des résidus.

Par ailleurs, les coefficients de régression doivent être interprétables. Nous pouvons mesurer la contribution de la variable $\mathbf{x}_j$ à la construction de la variable $\mathbf{y}$ à l'aide du coefficient de régression. Les coefficients sont donc très faciles à interpréter par :

$$\mathbf{y} = c_1 w_{11}\mathbf{x}_1 + \cdots + c_1 w_{1p}\mathbf{x}_p + \mathbf{y}_1. \tag{3.8}$$

Si le pouvoir explicatif de cette régression est faible, nous cherchons à construire une deuxième composantes $\mathbf{t}_2$. La composante $\mathbf{t}_2$ est une combinaison linéaire des $\mathbf{x}_j$ qui est non corrélée à $\mathbf{t}_1$

et explique bien le résidu $\mathbf{y}$. Nous obtenons $\mathbf{t}_2$ à l'aide de la formule :

$$\mathbf{t}_2 = w_{21}\mathbf{x}_{11} + \cdots + w_{2p}\mathbf{x}_{1p}, \tag{3.9}$$

où

$$w_{2j} = \frac{cov(\mathbf{x}_{1j}, \mathbf{y}_1)}{\sqrt{\sum_{j=1}^{p} cov^2(\mathbf{x}_{1j}, \mathbf{y}_1)}}, \tag{3.10}$$

et $\mathbf{x}_{1j}$ sont les résidus des régressions des variables $\mathbf{x}_j$ sur la composante $\mathbf{t}_1$. Nous effectuons ensuite une régression de $\mathbf{y}$ sur $\mathbf{t}_1$ et $\mathbf{t}_2$:

$$\mathbf{y} = c_1\mathbf{t}_1 + c_2\mathbf{t}_2 + \mathbf{y}_2. \tag{3.11}$$

Cette procédure est répétée et continue en utilisant de la même manière les résidus $\mathbf{y}_2, \mathbf{x}_{21}, \ldots, \mathbf{x}_{2p}$ des régression de $\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_p$ sur $\mathbf{t}_1$, $\mathbf{t}_2$.

Soit $\mathbf{X}$ une matrice de rang H. Nous avons choisi la même présentation que Tenenhaus dans (Tenenhaus, 1998) pour l'algorithme de régression *PLS*1 qui permet si besoin la prise en compte de données manquantes selon le principe de l'algorithme *NIPALS* :

1. Initialisation : $\mathbf{X}_0 = \mathbf{X}; \mathbf{y}_0 = \mathbf{y}$.
2. Itération : pour $h = 1, 2, \ldots, H$
   (a) $\mathbf{w}_h = \mathbf{X}'_{h-1}\mathbf{y}_{h-1}/\mathbf{y}'_{h-1}\mathbf{y}_{h-1}$
   (b) normer $\mathbf{w}_h$ à 1
   (c) $\mathbf{t}_h = \mathbf{X}_{h-1}\mathbf{w}_h/\mathbf{w}'_h\mathbf{w}_h$
   (d) $\mathbf{p}_h = \mathbf{X}'_{h-1}\mathbf{t}_h/\mathbf{t}'_h\mathbf{t}_h$
   (e) $c_h = \mathbf{y}'_{h-1}\mathbf{t}_h/\mathbf{t}'_h\mathbf{t}_h$
   (f) $\mathbf{X}_h = \mathbf{X}_{h-1} - \mathbf{t}_h\mathbf{p}'_h$ (déflation de $\mathbf{X}$)
   (g) $\mathbf{y}_h = \mathbf{y}_{h-1} - c_h\mathbf{t}_h$ (déflation de $\mathbf{y}$).

Ainsi, $\mathbf{X}_h$ est la matrice résiduelle de la régression linéaire multiple de $\mathbf{X}_{h-1}$ sur $\mathbf{t}_h$ et $\mathbf{y}_h$ est le vecteur des résidus de la régression linéaire simple de $\mathbf{y}_{h-1}$ sur $\mathbf{t}_h$.

Les coordonnées des vecteurs $\mathbf{w}_h$, $\mathbf{t}_h$, $\mathbf{p}_h$ et c_h représentent des pentes de droites des moindres carrés passant par l'origine. La coordonnée w_{hj} du vecteur $\mathbf{w}_h$ est le coefficient de régression de $\mathbf{y}_{h-1}$ dans la régression de la j-ième colonne de $\mathbf{X}_{h-1}$ sur la variable $\mathbf{y}_{h-1}$. La coordonnée t_{hi} du vecteur $\mathbf{t}_h$ est le coefficient de régression de $\mathbf{w}_h$ dans la régression sans constante de la variable définie par i-ième ligne de $\mathbf{X}_{h-1}$ sur la variable définie par le vecteur $\mathbf{w}_h$. La coordonnée p_{hj} du vecteur $\mathbf{p}_h$ est le coefficient de régression de $\mathbf{t}_h$ dans la régression de la j-ième colonne de $\mathbf{X}_{h-1}$ sur la variable $\mathbf{t}_h$. La valeur c_h est le coefficient de régression de $\mathbf{t}_h$ dans la régression de la variable $\mathbf{y}_{h-1}$ sur la variable $\mathbf{t}_h$.

En l'absence de donnée manquante, nous pouvons remplacer les étapes 2(a) et 2(b) par la formule suivante :

$$\mathbf{w}_h = \mathbf{X}'_{h-1}\mathbf{y}_{h-1} / \left\|\mathbf{X}'_{h-1}\mathbf{y}_h\right\|$$

et l'étape 2(c) peut alors être également remplacée par :

$$\mathbf{t}_h = \mathbf{X}_{h-1}\mathbf{w}_h$$

Propriétés de la régression *PLS*

Nous avons mis volontairement ce paragraphe pour présenter quelques propriétés mathématiques de la régression *PLS* au lecteur non initié même si aucune de ses propriétés nous servira pour la suite de cette thèse.

Les démonstrations des propriétés de l'algorithme de régression *PLS* lorsqu'il n'y a pas de données manquantes sont disponibles dans Tenenhaus (1998). Ces propriétés confirment l'orthogonalité recherchées et sont liées aux composantes. Ils sont décrits comme suit :

PROPOSITION 3.3.1.1 *Les vecteurs et les matrices issues de la régression PLS vérifient les propriétés suivantes :*

(a) $\boldsymbol{t}'_h\boldsymbol{t}_l = 0, \quad l > h$

(b) $\boldsymbol{w}'_h\boldsymbol{p}_h = 1$

(c) $\boldsymbol{w}'_h\boldsymbol{p}_l = 0, \quad l > h$

(d) $\boldsymbol{w}'_h\boldsymbol{w}_l = 0, \quad l > h$

(e) $\boldsymbol{t}'_h\boldsymbol{X}_l = 0, \quad l \geqslant h$

(f) $\boldsymbol{w}'_h\boldsymbol{X}'_l = 0, \quad l \geqslant h$

(g) $\boldsymbol{X}_h = \boldsymbol{X}\prod_{j=1}^{h}\left(\boldsymbol{I} - \boldsymbol{w}_j\boldsymbol{p}'_j\right)$, *pour* $h \geqslant 1$

La démonstration de chacune des propriétés se trouve dans Tenenhaus (1998), de la page 101 à 104.

Les vecteurs de poids étoiles, notés $\mathbf{w}^*_h$, sont définis à la propriété *(g)*. Nous rappelons que les composantes $\mathbf{t}_h$ sont définies à partir des résidus $\mathbf{X}_{h-1}$:

$$\mathbf{t}_h = \mathbf{X}_{h-1}\mathbf{w}_h.$$

En utilisant la propriété *(g)* de la Proposition 3.3.1.1, nous pouvons aussi exprimer les composantes en fonction de $\mathbf{X}$ (Tenenhaus, 1998) :

$$\mathbf{t}_h = \mathbf{X}\mathbf{w}^*_h.$$

où $\mathbf{w}_1^* = \mathbf{w}_1$. Et pour $h > 1$:

$$\mathbf{w}_h^* = \prod_{j=1}^{h-1} \left(\mathbf{I} - \mathbf{w}_j \mathbf{p}_j'\right) \mathbf{w}_h. \tag{3.12}$$

Nous décrivons les vecteurs poids « étoiles », $\mathbf{w}_h^*$, dans la Proposition 3.3.1.2.

PROPOSITION 3.3.1.2 *(Tenenhaus, 1998)*

(a) Les vecteurs $\mathbf{w}_h^*$ définis par la relation d'équation 3.12 vérifient l'équation de récurrence :

$$\mathbf{w}_h^* = \mathbf{w} - \mathbf{w}_{h-1}^* \boldsymbol{p}_h' \mathbf{w}, \tag{3.13}$$

avec $\boldsymbol{w}_1^* = \boldsymbol{w}_1$

(b) La matrice $\mathbf{W}_h^* = \left[\mathbf{w}_1^*, \ldots, \mathbf{w}_h^*\right]$ formée des vecteurs $\mathbf{w}_h^*$ définis par la relation d'équation notée 3.12 vérifie l'équation :

$$\mathbf{W}_h^* = \mathbf{W}_h (\boldsymbol{P}_h' \mathbf{W}_h)^{-1}. \tag{3.14}$$

Calcul des coefficients de régression *PLS*

Nous pouvons écrire la formule de régression de $\mathbf{y}$ sur les composantes $\mathbf{t}_1, \ldots, \mathbf{t}_H$ en fonction des variables $\mathbf{X}$. Notons $\mathbf{c}_H = (c_1, \ldots, c_H)'$ est le vecteur des coefficients de régression dans la régression de $\mathbf{y}$ sur $\mathbf{t}_1, \ldots, \mathbf{t}_H$. L'équation de régression *PLS* peut alors être écrite de la façon suivante :

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{T}_H \mathbf{c}_H \\ &= \mathbf{X} \mathbf{W}_H^* \mathbf{c}_H \\ &= \mathbf{X} \mathbf{W}_H (\mathbf{P}_H' \mathbf{W}_H)^{-1} \mathbf{c}_H. \end{aligned} \tag{3.15}$$

Ainsi, nous obtenons β_h qui est le vecteur des coefficients de régression dans la régression *PLS* de $\mathbf{y}$ sur $\mathbf{X}$ en utilisant h comme nombre de composantes de la régression *PLS*.

$$\begin{aligned} \beta_H &= \mathbf{W}_H^* \mathbf{c}_H \\ &= \mathbf{W}_H (\mathbf{P}_H' \mathbf{W}_H)^{-1} \mathbf{c}_H. \end{aligned} \tag{3.16}$$

3.3.2 Régression *PLS* en données incomplètes

La régression *PLS* permet la prise en compte des données manquantes selon le principe d'algorithme *NIPALS*. Cet algorithme est écrit sous la même forme que dans la section 3.3.1 mais avec quelques modifications. Les formules (3.5) et (3.6) sont modifiées de façon à n'utiliser que les données observées. Notons que $\mathbf{x}_{ij}$ est la valeur de la variable $\mathbf{x}_j$ pour l'observation i. Nous posons pour chaque observations i : (Tenenhaus, 1998) :

$$t_{1i} = \frac{\sum_{j:\, x_{ji} \text{ existe}} w_{1j}^{"} x_{ji}}{\sum_{j:\, x_{ji} \text{ existe}} (w_{1j})^2}, \tag{3.17}$$

où le coefficient w''_{1j} est défini de la façon suivante :

$$w'_{1j} = \frac{\sum_{i:\ x_{ji} \ et\ y_i \text{ existent}} x_{ji} y_i}{\sum_{i:\ x_{ji} \ et\ y_i \text{ existent}} y_i^2}, \tag{3.18}$$

par normalisation :

$$w''_{1j} = \frac{w'_{1j}}{\sqrt{\sum_{j=1}^{p} (w'_{1j})^2}}. \tag{3.19}$$

Les formules 3.17 et 3.18 correspondent à une application des principes de l'algorithme *NIPALS*. La coordonnée de vecteur $\mathbf{t}_{1i}$ dans la formule 3.17 représente la pente de la droite des moindres carrés, passant par l'origine, du nuage de points (w_{1j}, x_{ji}). De même façon à la coordonnée w'_{1j} dans la formule 3.18 représente la pente de la droite des moindres carrés, passant par l'origine, du nuage de points (y_i, x_{ji}).

Soit une matrice $\mathbf{X}$ de rang H, l'algorithme de la régression *PLS* selon le principe de l'algorithme *NIPALS* en cas de données incomplètes peut donc s'écrire de la façon suivante (Merola and Abraham, 2003) :

1. Initialisation : $\mathbf{X}_1 = \mathbf{X}, \mathbf{y}_1 = \mathbf{y}$
2. Itération : pour $h = 1, 2, \ldots, H$
   (a) $w_{j,h} = \frac{\sum_{\{i: x_{ij,h} \text{ et } y_{i,h} existe\}} x_{ij,h} y_{i,h}}{\sum_{\{i: x_{ij,h} \text{ et } y_{i,h} existe\}} y_{i,h}^2}$, $j = 1, \ldots, p$
   (b) normer $\mathbf{w}_h$ à 1
   (c) $t_{i,h} = \frac{\sum_{\{j: x_{ij,h} \ existent\}} x_{ij,h} w_{j,h}}{\sum_{\{j: x_{ij,h} \ existent\}} w_{j,h}^2}$, $i = 1, \ldots, n$
   (d) $p_{j,h} = \frac{\sum_{\{j: x_{ij,h} \ existent\}} x_{ij,h} t_{i,h}}{\sum_{\{j: x_{ij,h} \ existent\}} t_{i,h}^2}$, $j = 1, \ldots, p$
   (e) $c_h = \frac{\sum_{\{i: y_{i,h} \ existent\}} y_{i,h} t_{i,h}}{\sum_{\{i: y_{i,h} \ existent\}} t_{i,h}^2}$
   (f) $\mathbf{X}_{h+1} = \mathbf{X}_h - \mathbf{t}_h \mathbf{p}'_h$ pour $x_{ij,h}$ existant
   (g) $\mathbf{y}_{h+1} = \mathbf{y}_h - c_h \mathbf{t}_h$ pour $y_{i,h}$ existant

Chapitre 4

Choix du nombre de composantes

Dans le modèle de régression *PLS*, le nombre de composantes est généralement inconnu et l'un des objectifs de l'analyse de régression *PLS* est d'estimer ce nombre. La détermination du nombre de composantes de la régression *PLS* est une étape cruciale pour atteindre le modèle optimal, mais la sélection des composantes et des variables dans la construction des modèles *PLS* est encore sujet à débat (Meyer *et al.*, 2010). Si nous conservons trop peu de composantes, cela entraîne une perte d'informations. Par ailleurs, si nous choisissons trop de composantes, cette situation va entraîner le phénomène mieux connu d'*over-fitting*. L'*over-fitting* est une situation dans laquelle la réponse sera bien modélisée, mais malheureusement le modèle sera lié à de faibles performances prédictives car incluant du bruit dans la matrice des prédicteurs (Wiklund *et al.*, 2007).

Si le nombre de composantes sélectionnées dans la régression *PLS* est égal au nombre de variables (p) qui composent la matrice $\mathbf{X}$, les variables de la matrice $\mathbf{X}$ sont donc indépendantes. De plus, la régression *PLS* et la régression linéaire multiple donnent des résultats identiques. Ainsi, la régression *PLS* est une généralisation de la régression linéaire multiple.

Lorsque les jeux de données contiennent un grand nombre d'observations, ils sont généralement utilisés comme données d'entraînement ou *training set* en anglais avant toute analyse des «données de test» réelles (voir le figure 4.1). Augmenter la taille du jeu de données d'entraînement est nécessaire pour stabiliser le modèle de la régression *PLS*, ce qui conduit à des estimations non ambiguës du nombre optimal de composantes. Lorsque les jeux de données analysés sont de grande taille, ils peuvent généralement être divisés en deux parties : l'une contenant 2/3 des données pour l'entraînement du modèle et l'autre contenant 1/3 des données pour tester et valider le modèle et identifier le nombre optimal de composantes. Le choix de la taille de ces deux sous-parties peut avoir une grande influence sur la structure du modèle. Mais en règle générale, les données disponibles sont insuffisantes pour permettre de les fractionner en ensembles de données d'entraînement et de données de test, de sorte que d'autres méthodes dérivées de l'ensemble de données sont nécessaires.

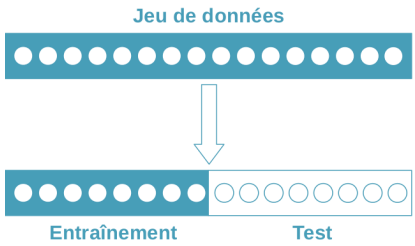


Figure 4.1 – Illustration de la séparation le jeu de données en un jeu d'entraînement et un jeu de test (Acazencott, 2020).

En d'autres termes, les mêmes données vont être utilisées à la fois pour d'entraînement et pour l'estimation des paramètres du modèle. Les données de test obtiennent les estimations de la réponse à partir du modèle de données d'entraînement. Idéalement, il faut alors disposer d'un jeu test supplémentaire et indépendant du jeu de données d'entraînement. Cette situation est relativement rare en pratique à cause de problématiques logistiques ou financières. Une solution à ce problème est la validation croisée qui est la technique la plus couramment utilisée pour contourner ce problème d'apprentissage (Wakeling and Morris, 1993).

Différents critères d'information ont été proposés pour tenter de corriger le biais du maximum de vraisemblance en ajoutant un terme de pénalité pour compenser l'*over-fitting* des modèles les plus complexes. Par exemple, le critère d'information d'Akaike, (en anglais *Akaike Information Criterion*, abrégé en *AIC*) et le critère d'information bayésien (en anglais *Bayesian Information Criterion*, abrégé en *BIC*).

4.1 Critères de la validation croisée sur le critère Q^2

La validation croisée est un moyen pratique et fiable de tester la signification prédictive de la régression *PLS*. Elle est devenue une méthode standard d'analyse de la régression *PLS*, et presque tous les logiciels de régression *PLS* fournissent cette méthode pour déterminer le nombre de composantes optimales de la régression *PLS*.

La validation croisée est le plus souvent réalisée en divisant les données en un certain nombre q de groupes, puis en construisant un certain nombre de modèles parallèles de données réduites d'un des groupes supprimés. Cette technique, baptisée validation croisée *q-Fold*, consiste donc à diviser le jeu initial de données en q groupes de taille identique (si possible) et de successivement en réserver un seul en tant que données de test. Ce processus est alors répété q fois afin que chaque groupe ait joué une fois le jeu de données de test (voir le figure 4.2). La validation croisée *q-Fold* est optimale lorsqu'elle est réalisée sur un faible nombre de sous-groupes, de l'ordre 5 à 10 (Shao, 1993; Wiklund *et al.*, 2007).

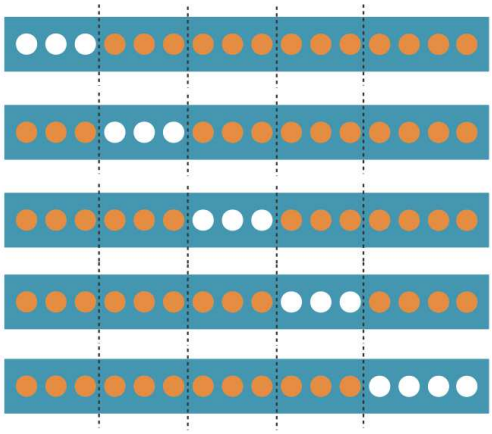


Figure 4.2 – Validation croisée à 5-*Fold* : Chaque point appartient à 1 des 5 jeux de test (en blanc) et aux 4 autres jeux d'entraînements (en orange) (Acazencott, 2020).

Si $q = n$, ce processus s'appelle *Leave-One-Out*, abrégé en *LOO*. La validation croisée *LOO*

consiste à ne sélectionner qu'un seul sujet comme la donnée de test, utilisant ainsi les $n-1$ données restantes afin d'établir le modèle. Ce procédé est alors répété n fois afin que chaque observation ait joué une fois le rôle du jeu de données de test (Gourvénec *et al.*, 2003).

Après avoir obtenu le modèle, nous avons la différence entre la valeur réelle de $\mathbf{y}$ et la valeur prédite de $\mathbf{y}$ ($\hat{\mathbf{y}}$). La somme des carrés de ces différences est calculée pour former une valeur du *PRedictive Error Sum of Squares*, abrégée en *PRESS*. La statistique du *PRESS* est un critère qui peut être utilisé pour évaluer la qualité de prédiction. Le *PRESS* s'écrit de la façon suivante :

$$PRESS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.1}$$

où n est nombre de données, y_i est la i-ème réponse et $\hat{y}_i$ est la réponse estimée à l'aide du modèle.

Le nombre de composantes $\mathbf{t}_1, \ldots, \mathbf{t}_H$ à retenir est habituellement déterminé par validation croisée. Pour chaque valeur h, nous calculons les prédictions $\hat{y}_{hi}$ et $\hat{y}_{h(-i)}$ de y_i à l'aide du modèle à h composantes, calculées en utilisant toutes les observations, puis sans utiliser l'observation i. Nous calculons alors le *PRESS* défini par :

$$PRESS_h = \sum_{i=1}^{n} \left(y_i - \hat{y}_{h(-i)}\right)^2. \tag{4.2}$$

La valeur Q_h^2 pour chaque composante $\mathbf{t}_H$ est calculée par :

$$Q_h^2 = 1 - \frac{PRESS_h}{RSS_{h-1}} \tag{4.3}$$

où $RSS_{h-1} = \sum_{i=1}^{n} \left(y_i - \hat{y}_{(h-1),i}\right)^2$ représente la somme des carrés résiduelle calculée avec le modèle à $h-1$ composantes. Pour $h = 1$, nous avons $RSS_0 = \sum_{i=1}^{n} (y_i - \bar{y})^2 = n-1$, puisque la variable $\mathbf{y}$ est centrée-réduite en utilisant la division par $n-1$ pour le calcul de la variance.

Une nouvelle composante $\mathbf{t}_h$ est significative si (Tenenhaus, 1998)

$$\sqrt{PRESS_h} \leqslant 0,95\sqrt{RSS_{h-1}} \Leftrightarrow Q_h^2 \geqslant 0,0975.$$

Dans le cas de données manquantes, Bertrand *et al.* (2014) ont développé le critère du Q^2-*LOO* qui est implémenté dans le *package* `plsRglm` du logiciel libre R. Lorsqu'il y a des jeux de données mixtes, c'est-à-dire que la matrice de $\mathbf{X}$ a des lignes complètes et incomplètes, nous pouvons calculer la prédiction de deux manières. La première façon est de prédire toutes les données en ligne en tant que données manquantes ; il s'agit du critère Q^2-*LOO standard*. La seconde façon est appelée critère Q^2-*LOO adaptative*. Ce critère prédit les valeurs séparément : les données absentes en tant que valeurs manquantes et les données non manquantes en tant que valeurs sans valeurs manquantes.

4.2 Critères d'information sur les critères *AIC* et *BIC*

Les critères *AIC* et *BIC* ont également été adaptés, ce qui leur permet de devenir des critères de choix de modèle, *i.e.* de sélection du nombre de composantes. Citons les applications de Li *et al.* (2002) qui ont utilisé le critère *AIC* et Krämer and Sugiyama (2011) qui ont développé le critère *BIC* sur la régression *PLS*. Notons que ces dernières applications ne concernent que la régression *PLS* usuelle en l'absence de données manquantes.

Le critère *AIC* est une mesure de la qualité d'un modèle statistique qui est proposée par Akaike (1974). Le *BIC* est un critère d'information dérivé d'*AIC* qui est proposé par Schwarz (1978). Ces critères sont généralement utilisés pour choisir le plus performant parmi plusieurs modèles de régression multiple ou parmi différents modèles de séries temporelles. Le modèle choisi est retenu sur la base d'une valeur minimale de l'*AIC* ou du *BIC*.

Nicole Krämer et Mikio L. Braun ont développé le concept de degrés de liberté (en anglais *degrees of freedom*, abrégé en *DoF*) du modèle de régression *PLS* qui a été présenté dans le *package* `dof` (Krämer and Braun, 2019). Comme nous utiliserons les *DoF* au chapitre suivant, nous pensons qu'il est important de rappeler quelques définitions à ce sujet.

La définition fondamentale des *DoF* provient de la structure des modèles linéaires qui peut être écrite par :

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}, \tag{4.4}$$

où $\mathbf{H}$ est appelée la matrice chapeau, ou en anglais *hat matrix*, qui permet de passer de la réponse observée à la réponse estimée. $\mathbf{H} \in \mathbb{R}^{p \times p}$ ne dépend pas de $\mathbf{y}$.

Dans le cas linéaire, les degrés de liberté sont définis comme la trace de la matrice chapeau (Hastie *et al.*, 2009) suivants :

$$DoF = \text{trace}(\mathbf{H}). \tag{4.5}$$

Dans le cas du modèle de la régression par les MCO, la matrice chapeau peux s'écrire comme suit :

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}', \tag{4.6}$$

où la trace de $\mathbf{H}$ est égale à son rang. Dans ce modèle, nous trouvons le résultat bien connu des *DoF* associés qui sont égaux à p.

La matrice chapeau associée dans le cadre de la régression *PLS* s'écrit alors :

$$\mathbf{H}_H = \mathbf{T}_H(\mathbf{T}_H'\mathbf{T}_H)^{-1}\mathbf{T}_H'. \tag{4.7}$$

Efron (2004) a donné la définition fondamentale des degrés de liberté pour les modèles linéaires généralisés. Sur cette base, Krämer and Sugiyama (2011) ont développé des degrés de liberté spécifiquement pour la régression *PLS*. Nous présentons le schéma de calcul des degrés de liberté dans la régression *PLS* à l'annexe A.

Les degrés de liberté dans la régression *PLS* incluent la limite inférieure du degré de liberté. Krämer et Sugiyama ont prouvé que la limite inférieure du degré de liberté de la première

composante est :

$$\widehat{DoF}(H=1) = 1 + \frac{\text{trace}(\mathbf{S})}{\lambda_{max}}, \tag{4.8}$$

où $\mathbf{S}$ est la matrice de corrélation de régression et λ_{max} est la plus grande valeur propre (Krämer and Sugiyama, 2011).

Les degrés de liberté de la régression *PLS* avec H composantes peut être défini de la façon suivante :

$$\widehat{DoF}(H) = 1 + \sum_{j=1}^{H} c_j \,\text{trace}(\mathbf{D}^j) - \sum_{j,l=1}^{H} \mathbf{t}_l' \mathbf{D}^j \mathbf{t}_l + (\mathbf{y} - \hat{\mathbf{y}}_H)' \sum_{j=1}^{H} \mathbf{D}^j \mathbf{v}_j + H \tag{4.9}$$

où $\mathbf{D} = \mathbf{X}\mathbf{X}'$, $\mathbf{V} = (\mathbf{v}_1, ..., \mathbf{v}_H) = \mathbf{T}(\mathbf{B}^{-1})'$, $\mathbf{B} = (\langle \mathbf{t}_i, \mathbf{D}^j \mathbf{y} \rangle)$, et $\mathbf{c} = \mathbf{B}^{-1}\mathbf{T}'\mathbf{y}$.

Krämer et Braun ont défini les critères *AIC* et *BIC* comme suit :

$$AIC = \frac{RSS}{n} + 2\frac{\widehat{DoF}}{n}\sigma^2 \tag{4.10}$$

$$BIC = \frac{RSS}{n} + \ln(n)\frac{\widehat{DoF}}{n}\sigma^2 \tag{4.11}$$

où *RSS* représente la somme des carrés résiduels associée à la régression *PLS*, n est le nombre d'observations, σ^2 est la variance inconnue des variables d'erreur et *DoF* est les degrés de liberté pour la régression *PLS* (Krämer and Sugiyama, 2011).

4.3 Sélection du nombre de composantes dans la régression *NIPALS-PLS* en pratique : le *package* `plsRglm` du logiciel

Le *package* `plsRglm` du logiciel (Bertrand *et al.*, 2014) fournit une analyse de la régression *NIPALS-PLS* de la réponse univariée et donne également l'analyse de la régression dans le cadre des modèles linéaires généralisés. Ce *package* fournit aussi certains critères pour déterminer le nombre de composantes et permet d'utiliser des données dans lesquelles il y a des structures de données manquantes sur les variables explicatives $\mathbf{X}$.

La fonction principale que nous avons implémentée est `plsR()`. Elle permet d'analyser le modèle de régression *NIPALS-PLS* avec certains critères pour déterminer le nombre de composantes, soit sur le jeu de données complet ou incomplet. Les critères sont *PRESS*, R^2, Q^2, *AIC*, et *BIC* avec ses dérivés, etc.

Troisième partie

Simulations, données réelles et applications

Chapitre 5

Plan de simulations

Plusieurs publications sur la régression *PLS* en données complètes utilisent des simulations pour analyser des aspects spécifiques du comportement de la régression *PLS*. Par exemple, Li *et al.* (2002) ont comparé, en se reposant sur des études par simulation, les critères *R-Wold* et *AIC* pour la sélection du nombre de composantes à inclure dans le modèle de la régression *PLS* afin qu'il constitue la base d'une représentation servant au contrôle statistique de processus multivariés.

Ce bref chapitre sur la simulation de la régression *NIPALS-PLS* ne prétend pas être exhaustif, mais illustre en quoi plusieurs propriétés peuvent être pertinentes pour les performances de la régression *NIPALS-PLS*, non seulement pour les données incomplètes, mais aussi pour les données complètes. De plus, l'interaction entre les critères et les méthodes utilisées peut être correctement identifiée pour déterminer le nombre de composantes.

5.1 Introduction

Le but principal de la simulation dans la régression *NIPALS-PLS* est de déterminer le nombre de composantes de la régression lorsque les données sont incomplètes. Cependant, nous nous sommes également intéressés à la performance de la régression *NIPALS-PLS* pour déterminer le nombre de composantes sur données complètes. La détermination du nombre de composantes a été effectuée en utilisant les différents critères. La question de l'identification du meilleur critère conduit donc à des questions plus profondes sur la sélection des composantes. Nous nous sommes posé les questions suivantes sur les performances globales de la régression *NIPALS-PLS* pour déterminer le nombre de composantes sans données manquantes (les données complètes) :

- Quelles sont les performances de ces critères sur la régression *NIPALS-PLS* ?
- Quelle est l'influence des différentes dimensions de données sur les performances de la régression $NIPALS - PLS$?
- Quelle est la durée de temps de calcul d'une régression *NIPALS-PLS* pour chaque dimension ?

Nous nous sommes également interrogés sur les performances de la régression *NIPALS-PLS* pour déterminer le nombre de composantes lorsqu'il y a des données manquantes (les données incomplètes) :

- Quelles sont les performances des différentes combinaison de critères et de méthodes sur la régression *NIPALS-PLS* ?
- Comment le type de mécanisme et la proportions de données manquantes affectent-ils les performances de la régression *NIPALS-PLS* ?
- Quelle est la différence de temps de calcul d'une régression *NIPALS-PLS* de chaque dimension, selon le type mécanisme de données manquantes et selon la méthode ?

Des réponses à ces questions sont fournies par les simulations dans le cas de données complètes et incomplètes en faisant varier les paramètres suivants :

1. les dimensions des jeux de données simulés,
2. le nombre vrai de composantes,
3. les critères de sélection de composantes,
4. les proportions de données manquantes,
5. les hypothèses sur les mécanismes des données manquantes,
6. les méthodes comparées.

L'objectif est alors d'estimer les performances des critères de sélection du nombre de composantes de la régression *NIPALS-PLS* que ce soit sur des données complètes (avec les paramètres 1, 2 et 3) ou sur des données incomplètes (avec tous les paramètres ci-dessus).

5.2 Cadre de travail : paramètres

Nombre d'observations (n) et nombre de variables explicatives (p)
Sept combinaisons sont formées à partir des dimensions de la matrice de données qui est soit de format vertical soit de format horizontal. Quand le nombre d'observations est supérieur au nombre de variables explicatives ($n > p$), la matrice est dite verticale. Dans le cas contraire, lorsque le nombre de variables explicatives est supérieur au nombre d'observations ($n < p$), la matrice est dite horizontale. Nous avons traité les sept situations (décrites ci-dessous) :

1. $n = 500$ et $p = 100$,
2. $n = 500$ et $p = 20$,
3. $n = 100$ et $p = 20$,

4. $n = 80$ et $p = 25$,
5. $n = 60$ et $p = 33$,
6. $n = 40$ et $p = 50$,
7. $n = 20$ et $p = 100$.

Nombre de variables réponses
Nous nous sommes limités à une réponse $\mathbf{y}$ univariée. Les données simulées de la réponse $\mathbf{y}$ sont également des données complètes.

Nombre de composantes (t)
Dans le cadre de la régression *NIPALS-PLS*, le point le plus important comme vu dans l'explication précédente est la détermination du nombre de composantes pouvant être choisies lors de la construction d'un modèle de régression *NIPALS-PLS*. Pour ce faire, nous avons fixé le nombre vrai de composantes de régression *NIPALS-PLS*, noté t^*, à deux, quatre et six pour chaque jeu de données. Après avoir effectué le processus de la simulation, nous comparons le nombre de composantes retenues de simulations avec t^* du modèle de régression *NIPALS-PLS* que nous avons initialement construit.

Proportion de données manquantes (d) et mécanismes des données manquantes
Nous avons simulé la proportion de données manquantes (d) dans des proportions de 5, 10, 15, 20, 25, 30, 35, 40, 45 et 50 %. Chacune de ces proportions de données manquantes est appliquée à chaque jeu de données. Ces proportions de données manquantes ont été simulées ensuite sur les variables explicatives $\mathbf{X}$ selon deux types de mécanismes de données manquantes : sous l'hypothèse *MCAR* et sous l'hypothèse *MAR*.

Paramétrages des données
Nous avons pris la formulation qui est proposée et décrite par Li *et al.* (2002). Elle est une généralisation de l'algorithme de Naes and Martens (1985). L'application de cette formule se trouve dans le *package* `plsRglm` qui est développé par Bertrand *et al.* (2014) lequel gère plusieurs types de distributions. En fait, ce *package* nous a donné plus de flexibilité dans nos recherches. De plus, ce *package* contient une fonction qui peut générer une valeur de réponse $\mathbf{y}$ et les variables explicatives $(\mathbf{x}_1, \ldots, \mathbf{x}_p)$ pour un modèle de régression *PLS* avec un nombre de composantes prédéfini. Cette fonction s'appelle `simul_data_UniYX`.

Les simulations ont été effectuées d'utiliser le package *plsRglm* en adaptant la fonction *simul_data_UniYX* Bertrand *et al.* (2014). Cette fonction produit une valeur de réponse $\mathbf{y}$ et les variables explicatives $(\mathbf{X}_1, \ldots, \mathbf{X}_p)$ qui est prise à partir d'un modèle avec un certain nombre de composantes. L'algorithme utilisé est celui décrit dans l'article de Li *et al.* (2002), qui est une généralisation de l'algorithme de Naes and Martens (1985).

Le modèle de simulations pour chaque t^* ($2 \leq t^* \leq 6$) a été développés par Bertrand *et al.* (2014).

La matrice $\mathbf{X}$ avec le nombre d'observations (n) sont :

$$\mathbf{X} = \sum_{h=1}^{t^*} \nu_h \zeta_h' + \xi \tag{5.1}$$

où $\xi = (\varepsilon_1, ..., \varepsilon_6)$ et $\varepsilon_a (a = 1,\ldots, 6)$ sont simulés en tant que les variables normales mutuellement indépendantes avec la valeur de moyenne = 0 et var(ε_a) = 0,01.
ν_h (h = 1, 2, 3, 4, 5, 6) est généré en tant que les variables normales mutuellement indépendantes avec la valeur de moyenne = 0 et var(ν_1) = 10, var(ν_2) = 8, var(ν_3) = 6, var(ν_4) = 4, var(ν_5) = 2 et (ν_6) = 0.5.
ζ_h est défini dans différentes valeurs en fonction du nombre de variables qui ont été développées dans le package *plsRglm* sur la fonction *simul_data_UniYX* (Bertrand *et al.*, 2014).

La matrice $\mathbf{Y}$ sont générées par :

$$\mathbf{Y} = \sum_{h=1}^{t^*} \mathbf{z}_h \eta_{t^* h}' + \varpi \tag{5.2}$$

où $\eta_{t^* h}$ sont définis par :

$$\begin{array}{lll}
\eta_{21} = [1,1,1]'/3^{1/2}, & \eta_{22} = [1,1,1]'/3^{1/2}, & \eta_{31} = [1,2,1]'/6^{1/2}, \\
\eta_{32} = [1,1,1]'/3^{1/2}, & \eta_{33} = [1,1,1]'/3^{1/2}, & \eta_{41} = [1,1,1,1]'/2, \\
\eta_{42} = [1,1,1,1]'/2, & \eta_{43} = [1,1,1,1]'/2, & \eta_{44} = [1,1,1,1]'/2, \\
\eta_{51} = [1,1,1,1]'/2, & \eta_{52} = [1,1,1,1]'/2, & \eta_{53} = [1,1,1,1]'/2, \\
\eta_{54} = [1,1,1,1]'/2, & \eta_{55} = [1,1,1,1]'/2, & \eta_{61} = [1,1,1,1]'/2, \\
\eta_{62} = [1,1,1,1]'/2, & \eta_{63} = [1,1,1,1]'/2, & \eta_{64} = [1,1,1,1]'/2, \\
\eta_{65} = [1,1,1,1]'/2, & \eta_{66} = [1,1,1,1]'/2. &
\end{array} \tag{5.3}$$

$\mathbf{z}_h = \nu_h + \mathbf{f}_h$ où $\mathbf{f}_h$ (h = 1, 2, 3, 4, 5, 6) est généré sous forme les variables normales indépendantes avec la valeur de moyenne = 0 et var(ς_1) = 0,25, var(ς_2) = 0,125, var(ς_3) = 0,05, var(ς_4) = 0,0125, var(ς_5) = 0,005 et var(ς_6) = 0,00125.
$\varpi = (\varpi_1, \ldots, \varpi_{t^*})$ est généré par un vecteur aléatoire avec une distribution normale multivariée $\mathcal{N}(\mathbf{0}, \mathbf{S})$ où $\mathbf{S} = \sigma^2 \left[(1-\lambda)\mathbf{I} + \lambda \mathbf{1}\mathbf{1}'\right]$ avec $\sigma^2 = 0.001$, $\lambda = 0.6$, $\mathbf{I}$ est une matrice d'identité et $\mathbf{1}$ est un vecteur d'unité.
Donc, les composantes $\mathbf{t}_h$ sont approximativement égaux à ν_h ($h = 1, \ldots, t^*$) et la variable réponse $\mathbf{y}$ dépend essentiellement de ν_h ($h = 1, \ldots, t^*$) plus le bruit (Li *et al.*, 2002).

Méthodes
La comparaison des résultats obtenus s'effectue entre la régression *NIPALS-PLS* sur des données incomplètes et la régression *NIPALS-PLS* sur de données imputées que les méthodes d'imputation sont *MICE*, *KNNimpute* et *SVDimpute*. Nous disons « la méthode » pour la rendre plus facile en indiquant la combinaison entre la régression *NIPALS-PLS* et les méthodes d'imputation.

La méthode *NIPALS-PLS* fournit seulement la régression *NIPALS-PLS* sur les données incomplètes. La méthode *MICE*, la méthode *KNNimpute* et la méthode *SVDimpute* illustrent la régression *NIPALS-PLS* sur les données imputées qui prennent l'imputation *MICE*, *KNNimpute*,

et *SVDimpute* respectivement. Pour plus de clarté et de simplicité, nous présentons les méthodes dans l'image ci-dessous.

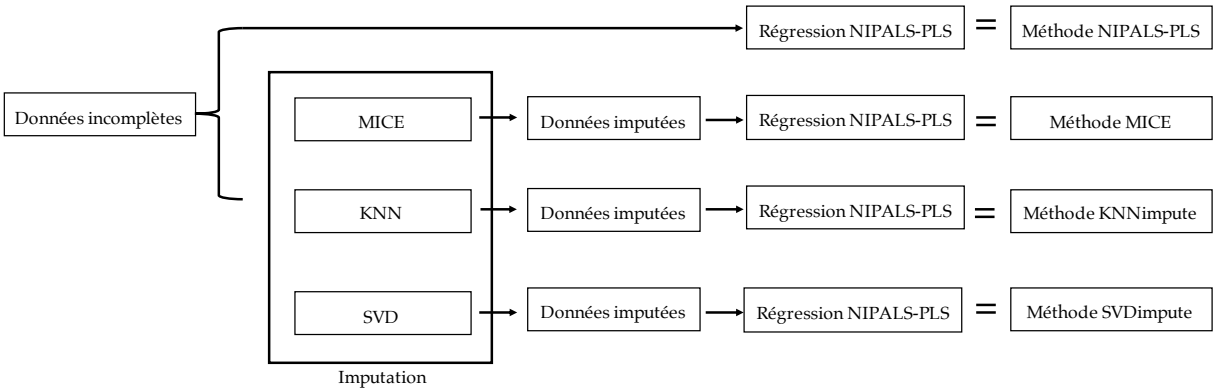


Figure 5.1 – Méthodes.

Critères
Nous avons choisi six critères : Q^2-*LOO*, Q^2-10-*Fold*, *AIC*, *AIC-DoF*, *BIC* et *BIC-DoF*.

Ressources matérielles de simulations
Nous avons travaillé sur l'ordinateur avec 2 cœurs et 1 processeur *Dual-Core Intel Core i5* et les deux serveurs de l'université de Strasbourg avec 8 cœurs et 64 processeurs *AMD opteron* et 28 cœurs et 56 processeurs *Intel Xeon*.

5.3 Processus de simulations

Le processus de simulations dans cette thèse est décrit comme suit :

1. Les données complètes ont été simulées d'après Li *et al.* (2002) avec différentes dimensions des jeux de données et les t^*.
   À partir de ces données complètes, nous calculons les composantes optimales (composantes retenues) en utilisant six critères. Nous obtiendrons donc les résultats de la régression *NIPALS-PLS* dans le cas de données complètes jusqu'à ce processus.

2. Les données manquantes sont créées sous l'hypothèse MCAR ou sous l'hypothèse MAR avec la proportion de données manquantes, par exemple 5%.

3. Les valeurs manquantes sont imputées en utilisant l'imputation *MICE*, *KNNimpute* et *SVDimpute*. Nous avons alors les données imputées ; le nombre de composantes est ensuite choisi sur les données complètes, après imputation, en utilisant la régression *NIPALS-PLS*.

4. Le nombre de composantes est choisi à l'aide six critères.
   Le critère Q^2-*LOO* sur les données incomplètes adopte la validation croisée *LOO* par *standard* et par *adaptative*

5. Nous avons fixé le nombre maximal de composantes pouvant être extraites à huit composantes. Notons que le nombre vrai de composantes est de 2, 4 ou 6 composantes.

6. Pour chaque combinaison du nombre vrai de composantes, de la proportion de données manquantes et du mécanisme supposé d'apparition des données manquantes, 1000 réplications ont été réalisées.

À partir de ces 1000 simulations, nous comptons finalement le nombre de composantes retenues (c'est-à-dire que le nombre de fois que le bon nombre de composantes est sélectionné) qui correspondent au t^* déterminé. Ce processus de simulations est illustré sur la Figure 9.1 dans la Section 9.3.4.

Chapitre 6

Résultats des simulations pour données complètes

Les Tableaux 6.1 à 6.5 donnent le nombre de composantes retenues, sur un total de 1000 simulations, pour chaque critère sur les sept cas dans les cas de données complètes. Les résultats sont exprimés en nombre de composantes retenues en fonction du nombre vrai de composantes (t^*) qui prend successivement les valeurs 2, 4 et 6. Le nombre moyen de composantes retenues est aussi indiqué.

La comparaison des performances des critères pour déterminer le nombre de composantes pour la régression *NIPALS-PLS* est présentée dans la première section (voir la Section 6.1), en fonction du nombre de composantes et en fonction des dimensions (tableau de forme verticale ou tableau de forme horizontale). La deuxième section de ce chapitre fournit l'influence des différentes dimensions des jeux de données (voir la Section 6.2). La troisième section répertorie la durée de temps de calculs pour chaque dimension de jeu de données et chaque nombre vrai de composantes (voir la Section 6.3). À la fin de ce chapitre (voir la Section 6.4), nous fournissons une brève conclusion sur la détermination du nombre de composantes sur la régression *NIPALS-PLS* en présence de données complètes.

6.1 Comparaison de différents critères

6.1.1 Données sous forme de matrice verticale

Les Tableaux 6.1 à 6.3 montrent la fréquence du nombre de composantes retenues pour des matrices verticales. Les résultats montrent que les critères Q^2-*LOO* et Q^2-10-*Fold* sont globalement les meilleurs pour déterminer le nombre vrai de composantes pour la régression *NIPALS-PLS*. Ils choisissent le nombre vrai de composantes presque parfaitement sur 1000 simulations. Les moyennes du nombre de composantes retenues sont presque les mêmes que $t^* = 2$, 4 et 6 (voir lignes 1 et 2 pour chaque t^*).

Les résultats du nombre de composantes retenues par les critères *AIC* et *BIC* montrent qu'ils

ont de moins bonnes performances. Dans la majorité des cas où la matrice est verticale, les performances de ces critères tendent à sélectionner un nombre trop grand de composantes (voir lignes 3 et 5 des tableaux pour chaque t^*).

Sur les critères *AIC-DoF* et *BIC-DoF*, l'augmentation du nombre de variables et la diminution du nombre d'observations permettront des prévisions qui sont moins précises que le nombre réel de composantes. Les critères *AIC-DoF* et *BIC-DoF* ont une bonne performance, comme le critère Q^2, lorsque le nombre d'observations est très supérieur au nombre de variables, par exemple ici lorsque $n = 500$ et $p = 100$ ou $n = 500$ et $p = 20$.

Par exemple, les moyennes du nombre de composantes du critère *BIC-DoF* pour $n = 500$ et $p = 20$ est 2,13 et pour $n = 500$ et $p = 100$ est 2,28, lorsque le cas de $t^* = 2$. Un autre exemple, lorsque le cas de $t^* = 2$ et $n = 500$ et $p = 20$, les moyennes du nombre de composantes du critère *BIC-DoF* est 2,13 et pour $n = 500$ et $n = 100$ est 2,23. Cela signifie que l'augmentation du nombre de variables et la diminution du nombre d'observations permettront des prévisions sur le critère *BIC-DoF* moins précises du nombre vrai de composantes.

Table 6.1 – Nombre de composantes retenues pour $n = 500$ et $p = 100$ selon les critères et le nombre vrai de composantes (t^*) sur 1000 simulations

t^*	Critères	Nombre de composantes retenues								Moyenne du nombre de composantes
		1	2	3	4	5	6	7	8	
2	Q^2-*LOO*	0	1000	0	0	0	0	0	0	2,00
	Q^2-10-*Fold*	0	1000	0	0	0	0	0	0	2,00
	AIC	0	0	0	2	986	12	0	0	5,01
	AIC-DoF	0	823	0	52	55	46	21	3	2,58
	BIC	0	0	0	897	103	0	0	0	4,10
	BIC-DoF	0	906	0	41	25	19	9	0	2,28
4	Q^2-*LOO*	0	0	0	1000	0	0	0	0	4,00
	Q^2-10-*Fold*	0	0	0	1000	0	0	0	0	4,00
	AIC	0	0	0	0	0	11	979	10	7,00
	AIC-DoF	0	0	0	699	0	92	67	142	4,95
	BIC	0	0	0	0	0	906	94	0	6,09
	BIC-DoF	0	0	0	811	0	72	46	71	4,57
6	Q^2-*LOO*	0	0	0	0	0	1000	0	0	6,00
	Q^2-10-*Fold*	0	0	0	0	0	1000	0	0	6,00
	AIC	0	0	0	0	0	0	0	1000	8,00
	AIC-DoF	0	0	0	0	0	496	168	336	6,84
	BIC	0	0	0	0	0	0	0	1000	8,00
	BIC-DoF	0	0	0	0	0	726	51	223	6,50

Table 6.2 – Nombre de composantes retenues pour $n = 500$ et $n = 100$ selon les critères et le nombre vrai de composantes (t^*) sur 1000 simulations

(a) $n = 500$ et $p = 20$

t^*	Critères	Nombre de composantes retenues								Moyenne du nombre de composantes
		1	2	3	4	5	6	7	8	
2	Q^2-LOO	0	1000	0	0	0	0	0	0	2,00
	Q^2-10-Fold	0	1000	0	0	0	0	0	0	2,00
	AIC	0	0	888	112	0	0	0	0	3,11
	AIC-DoF	0	874	2	96	21	3	1	3	2,29
	BIC	0	7	993	0	0	0	0	0	2,99
	BIC-DoF	0	942	0	49	7	2	0	0	2,13
4	Q^2-LOO	0	0	0	1000	0	0	0	0	4,00
	Q^2-10-Fold	0	0	0	1000	0	0	0	0	4,00
	AIC	0	0	0	0	991	9	0	0	5,01
	AIC-DoF	0	0	0	825	9	116	41	9	4,40
	BIC	0	0	0	10	990	0	0	0	4,99
	BIC-DoF	0	0	0	907	0	68	22	3	4,21
5	Q^2-LOO	0	0	0	0	0	1000	0	0	6,00
	Q^2-10-Fold	0	0	0	0	0	1000	0	0	6,00
	AIC	0	0	0	0	0	0	1000	0	7,00
	AIC-DoF	0	0	0	0	0	127	718	155	7,03
	BIC	0	0	0	0	0	46	954	0	6,95
	BIC-DoF	0	0	0	0	0	415	500	85	6,67

(b) $n = 100$ et $p = 20$

t^*	Critères	Nombre de composantes retenues								Moyenne du nombre de composantes
		1	2	3	4	5	6	7	8	
2	Q^2-*LOO*	1	993	6	0	0	0	0	0	2,01
	Q^2-10-*Fold*	2	996	2	0	0	0	0	0	2,00
	AIC	0	0	191	797	12	0	0	0	3,82
	AIC-DoF	0	775	8	77	87	27	17	9	2,67
	BIC	0	0	770	229	1	0	0	0	3,23
	BIC-DoF	0	909	0	54	29	5	3	0	2,23
4	Q^2-*LOO*	0	0	3	995	2	0	0	0	4,00
	Q^2-10-*Fold*	0	0	4	996	0	0	0	0	4,00
	AIC	0	0	0	0	391	604	5	0	5,61
	AIC-DoF	0	0	0	679	13	135	131	42	4,84
	BIC	0	0	0	2	862	136	0	0	5,13
	BIC-DoF	0	0	0	821	0	94	68	17	4,46
6	Q^2-*LOO*	0	0	0	0	0	995	5	0	6,01
	Q^2-10-*Fold*	0	0	1	0	0	999	0	0	6,00
	AIC	0	0	0	0	0	0	583	417	7,42
	AIC-DoF	0	0	0	0	0	409	237	354	6,95
	BIC	0	0	0	0	0	4	912	84	7,08
	BIC-DoF	0	0	0	0	0	568	205	227	6,66

Table 6.3 – Nombre de composantes retenues pour $n = 80$ et $n = 60$ selon les critères et le nombre vrai de composantes (t^*) sur 1000 simulations

(a) $n = 80$ et $p = 25$

t^*	Critères	Nombre de composantes retenues								Moyenne du nombre de composantes
		1	2	3	4	5	6	7	8	
2	Q^2-*LOO*	0	996	4	0	0	0	0	0	2,00
	Q^2 10-*Fold*	1	999	0	0	0	0	0	0	2,00
	AIC	0	0	18	620	351	11	0	0	4,36
	AIC-DoF	0	697	5	59	67	68	51	53	3,17
	BIC	0	0	213	734	53	0	0	0	3,84
	BIC-DoF	0	895	0	45	32	16	7	5	2,32
4	Q^2-*LOO*	0	0	5	983	12	0	0	0	4,01
	Q^2-10-*Fold*	0	0	16	984	0	0	0	0	3,98
	AIC	0	0	0	0	42	799	159	0	6,12
	AIC-DoF	0	0	0	636	13	109	122	120	5,08
	BIC	0	0	0	0	316	666	18	0	5,70
	BIC-DoF	0	0	0	803	0	82	69	46	4,56
6	Q^2-*LOO*	0	0	0	2	0	990	8	0	6,00
	Q^2-10-*Fold*	0	1	8	3	0	988	0	0	5,97
	AIC	0	0	0	0	0	0	76	924	7,92
	AIC-DoF	0	0	0	0	0	473	130	397	6,92
	BIC	0	0	0	0	0	0	403	597	7,60
	BIC-DoF	0	0	0	0	0	656	77	267	6,61

(b) $n = 60$ et $p = 33$

t^*	Critères	Nombre de composantes retenues								Moyenne du nombre de composantes
		1	2	3	4	5	6	7	8	
2	Q^2-*LOO*	0	969	31	0	0	0	0	0	2,03
	Q^2-10-*Fold*	0	996	4	0	0	0	0	0	2,00
	AIC	0	0	0	24	302	420	200	54	5,96
	AIC-DoF	0	484	5	26	40	57	81	307	4,65
	BIC	0	0	6	210	508	226	45	5	5,11
	BIC-DoF	0	800	0	26	35	43	28	68	2,88
4	Q^2-*LOO*	0	0	21	945	34	0	0	0	4,01
	Q^2-10-*Fold*	0	0	38	960	2	0	0	0	3,96
	AIC	0	0	0	0	0	51	401	548	7,50
	AIC-DoF	0	0	0	485	10	79	84	342	5,79
	BIC	0	0	0	0	3	233	544	220	6,98
	BIC-DoF	0	0	1	674	0	75	79	171	5,07
6	Q^2-*LOO*	0	0	1	6	0	951	41	0	6,02
	Q^2-10-*Fold*	0	9	38	10	0	942	0	0	5,82
	AIC	0	0	0	0	0	0	0	999	7,99
	AIC-DoF	0	0	0	0	0	462	96	441	6,97
	BIC	0	0	0	0	0	0	3	996	7,99
	BIC-DoF	0	0	0	0	0	636	72	291	6,65

6.1.2 Données sous forme de matrice horizontale

Les Tableaux 6.4 et 6.5 résument la fréquence du nombre de composantes retenues sur les dimensions horizontales ($n = 20$ et $n = 40$). En général, les performances du critère Q^2, soit Q^2-LOO ou Q^2-10-*Fold*, sont également les meilleurs critères pour déterminer le nombre vrai de composantes sur la régression *NIPALS-PLS* sauf lorsque $n = 20$ et $t^* = 6$. Ces critères déterminent le nombre correct de composantes qui est supérieur aux trois quarts des simulations sur 1000 simulations (voir lignes 1 et 2 du Tableau **??**).

Sur 1000 simulations, tous les nombres de composantes retenues des critères *AIC* et *BIC* sont loin de la situation du nombre vrai de composantes. Dans la grande majorité des situations, le nombre de composantes retenues est en moyenne plus grand que le nombre réel. Autrement dit, leurs performances sont très mauvaises. Par exemple dans le cas $n = 40$ et $t^* = 2$, ces critères sélectionnent huit composantes dans 100% de cas. À savoir que huit composantes est le nombre de composantes maximale pouvant être extraites du processus de simulations (voir lignes 3 et 5 le tableau **??**(a)).

Les performances des critères *AIC-DoF* et *BIC-DoF* donnent également les mêmes performances avec les critères *AIC* et *BIC*. Leurs performances tendent à sélectionner un nombre trop grand de composantes.

Table 6.4 – Nombre de composantes retenues pour $n = 40$ et $p = 50$ selon les critères et le nombre vrai de composantes (t^*) sur 1000 simulations

t^*	Critères	Nombre de composantes retenues								Moyenne du nombre de composantes
		1	2	3	4	5	6	7	8	
2	Q^2-*LOO*	3	908	89	0	0	0	0	0	2,09
	Q^2-10-*Fold*	4	990	6	0	0	0	0	0	2,00
	AIC	0	0	0	0	0	0	0	1000	8,00
	AIC-DoF	0	151	12	23	26	14	15	759	6,82
	BIC	0	0	0	0	0	0	0	1000	8,00
	BIC-DoF	0	258	0	22	29	13	17	661	6,23
4	Q^2-*LOO*	0	0	44	799	157	0	0	0	4,11
	Q^2-10-*Fold*	0	2	85	908	5	0	0	0	3,92
	AIC	0	0	0	0	0	0	0	1000	8,00
	AIC-DoF	0	0	0	278	24	55	63	580	6,64
	BIC	0	0	0	0	0	0	0	1000	8,00
	BIC-DoF	0	0	0	357	2	61	65	515	6,38
6	Q^2-*LOO*	0	2	2	7	0	770	216	0	6,17
	Q^2 10-*Fold*	3	27	148	16	4	801	0	0	5,39
	AIC	0	0	0	0	0	0	0	999	7,99
	AIC-DoF	0	0	0	0	0	394	67	538	7,14
	BIC	0	0	0	0	0	0	0	999	7,99
	BIC-DoF	0	0	0	0	0	514	39	446	6,93

Table 6.5 – Nombre de composantes retenues pour et $n = 20$ et $p = 100$ selon les critères et le nombre vrai de composantes (t^*) sur 1000 simulations

t^*	Critères	Nombre de composantes retenues								Moyenne du nombre de composantes
		1	2	3	4	5	6	7	8	
2	Q^2-*LOO*	6	475	316	90	31	13	5	0	2,53
	Q^2-10-*Fold*	15	974	11	0	0	0	0	0	2,00
	AIC	0	0	0	0	0	0	0	1000	8,00
	AIC-DoF	0	0	0	0	0	0	1	999	8,00
	BIC	0	0	0	0	0	0	0	1000	8,00
	BIC-DoF	0	0	0	0	0	1	0	999	8,00
4	Q^2-*LOO*	0	0	3	995	2	0	0	0	4,00
	Q^2-10-*Fold*	0	0	4	996	0	0	0	0	4,00
	AIC	0	0	0	0	391	604	5	0	5,61
	AIC-DoF	0	0	0	679	13	135	131	42	4,84
	BIC	0	0	0	2	862	136	0	0	5,13
	BIC-DoF	0	0	0	821	0	94	68	17	4,46
6	Q^2-LOO	16	52	38	23	6	49	61	0	1,08
	Q^2-10-Fold	116	314	299	62	71	138	0	0	3,07
	AIC	0	0	0	0	0	0	0	1000	8,00
	AIC-DoF	0	0	0	0	0	0	9	991	7,99
	BIC	0	0	0	0	0	0	0	1000	8,00
	BIC-DoF	0	0	0	0	0	1	10	989	7,99

6.2 Comparaison de jeux de données de différentes dimensions

De manière générale, les dimensions du jeu de données n'influent pas sur les performances des critères Q^2-*LOO* et Q^2-10-*Fold*. Au contraire, les dimensions du jeu de données influent sur les performances des critères *AIC-DoF* et *BIC-DoF*. Le nombre de composantes retenues diminue avec l'augmentation du nombre de variables et la diminution du nombre d'observations.

En effet, les dimensions du jeu de données, que ce soit sur les matrices verticales ou sur les matrices horizontales ne modifie pas généralement notablement les performances des critères *AIC* et *BIC*. Leurs performances tendent globalement à sélectionner un nombre trop grand de composantes quelle que soit la dimension des données.

Ainsi, lorsque le nombre d'observations est supérieur au nombre de variables, cela permettra des prévisions utilisant le critère Q^2 plus précises pour choisir le nombre vrai de composantes.

6.3 Comparaison des différents temps calculs

Le Tableau 6.6 présente la moyenne des temps d'exécution d'une simulation (en secondes) qui est calculée sur 1000 simulations sur de données complètes. Le temps de calcul dépend globalement

de la dimension du jeu de données. La moyenne de temps d'exécution d'une simulation (en secondes) qui est calculée sur 1000 simulations augmente avec le nombre d'observations et le nombre de variables.

Table 6.6 – Moyenne de temps d'exécution d'une simulation (en secondes) calculée sur 1000 simulations sur de données complètes

t^*	Dimension de données						
	$n = 500$ et $p = 100$	$n = 500$ et $p = 20$	$n = 100$ et $p = 20$	$n = 80$ et $p = 25$	$n = 60$ et $p = 33$	$n = 50$ et $p = 40$	$n = 20$ et $p = 100$
2	12,6130	3,2625	2,5804	2,4288	1,6927	1,4656	1,3293
4	12,4906	3,2064	2,4357	2,3938	1,3653	1,3624	1,5057
6	12,6757	3,2739	3,2216	2,8888	2,4363	1,9031	1,5506

6.4 Conclusion de simulations en données complètes

Une comparaison a été effectuée pour les six critères : Q^2-*LOO*, Q^2-10-*Fold*, *AIC*, *AIC-DoF*, *BIC* et *BIC-DoF* sur la détermination du nombre de composantes à inclure dans une régression *PLS* sous différentes dimensions pour des jeux de données $n > p$, (matrice verticale) et $n < p$ (matrice horizontale). Nous résumons ici les résultats des simulations qui donnent une estimation du nombre de composantes plus proche du nombre vrai de composantes en données complètes.

1. Les critères Q^2-*LOO* et Q^2-10-*Fold* sont les meilleurs critères et donnent systématiquement des pourcentages de choix corrects plus élevés que ceux obtenus par d'autres critères, lorsque les dimensions de données sont verticales. Ces critères atteignent globalement les pourcentages presque parfaits pour déterminer le nombre de vraies composantes quelle que soit la dimension de données.

2. Les critères *AIC* et *BIC* tendent à sélectionner un nombre trop grand de composantes quelle que soit la dimension des tableaux de données.

3. Le critère *BIC-DoF* a une bonne performance, comme le critère Q^2, lorsque le nombre d'observations est très supérieur au nombre de variables, par exemple dans cette thèse $n = 500$ et $p = 100$ ou $n = 500$ et $p = 20$.

4. Le temps de calcul augmente avec le nombre d'observations et le nombre de variables.

Chapitre 7

Résultats des simulations pour données incomplètes

Les figures dans les Sections B.1 et B.2 de l'Annexe B présentent les performances de chaque critère avec quatre méthodes, en fonction de la proportion de données manquantes (de 5% à 50% par incrément de 5%) pour chaque nombre vrai de composantes et chaque dimension des tableaux de données. Les résultats sont exprimés en nombre de simulations, sur un total de 1000, pour lesquelles le nombre retenu de composantes est égal au nombre vrai de composantes (t^*). t^* a été pris successivement égal à 2, 4 et 6. Nous présentons également la comparaison des performances pour différents mécanismes d'apparition de données manquantes (sous l'hypothèse *MCAR* et sous l'hypothèse *MAR*).

La comparaison des performances de chaque critère pour la détermination du nombre de composantes de la régression *NIPALS-PLS* est présentée dans la première section pour chaque nombre vrai de composantes et soit sur les matrices de forme verticale soit sur les matrices de forme horizontale (voir la Section 7.1). La deuxième section de ce chapitre fournit également une comparaison des performances en fonction des mécanismes d'apparition et de la proportion de données manquantes (voir la Section 7.2). Le temps de calcul est présenté dans la troisième section (voir la Section 7.3) pour chacune des tailles des jeux de données, chaque hypothèse de données manquantes, chaque méthode et chaque nombre vrai de composantes. La dernière section présente une conclusion de l'étude sur la détermination du nombre de composantes de la régression *NIPALS-PLS* en présence de données incomplètes (voir la Section 7.4).

7.1 Comparaison de différents critères et de méthodes

7.1.1 Tableaux de données de forme verticale

Les figures dans l'Annexe B.1 représentent le nombre de simulations, sur un total de 1000, pour lesquelles le nombre retenu de composantes est égal au nombre vrai de composantes, en fonction de chaque critère, chaque dimension de données et de la proportion de données manquantes (de 5% à 50% par 5%) aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*. La

performance des critères : Q^2-*LOO*, Q^2-10-*Fold*, *AIC-DoF* et *BIC-DoF* dans la détermination du nombre vrai de composantes augmente généralement à mesure que le nombre d'observations augmente et, comme anticipé, diminue lorsque la proportion de données manquantes augmente, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*. Lorsque le nombre d'observations est très supérieur au nombre de variables ($n >> p$), par exemple dans cette thèse pour $n = 500$ et $p = 100$ ou pour $n = 500$ et $p = 20$, les performances des critères pour déterminer le nombre de composantes sont très variables les unes des autres. Les détails des performances des critères et des méthodes sont disponibles ci-dessous.

Nombre vrai de composantes égal à 2

- **Q^2-LOO**
  Comparativement à d'autres méthodes, la méthode *NIPALS-PLS* avec le critère Q^2-*LOO* *standard* ou *adaptative* sous l'hypothèse *MCAR* fournit globalement une performance satisfaisante avec n'importe quelle proportion de données manquantes. Le nombre de composantes retenues est presque parfait pour toutes les proportions de données manquantes lorsque le nombre d'observations augmente (voir la Figure B.1).
  Par exemple, les performances de la méthode *NIPALS-PLS* avec le critère Q^2-*LOO* *standard* sont très bonnes puisque le nombre de fois que le correct nombre de composantes est choisi, en moyenne, dans plus de 96,04% des cas lorsque les dimensions de données sont verticales.
  La méthode *KNNimpute* est aussi une très bonne méthode et donne également de bons résultats pour le choix du nombre de composantes aussi bien sous une hypothèse *MCAR* que *MAR*. Ses performances diminuent généralement avec l'augmentation de la proportion de données manquantes.

  Les résultats dans le cas des dimensions de données pour lesquelles $n >> p$ sont intéressants. Les performances de la méthode *NIPALS-PLS* pour le critère Q^2-*LOO*, aussi bien *standard* qu'*adaptative*, sont parfaites quelle que soit la proportion de données manquantes sous l'hypothèse *MCAR*. Le nombre de composantes retenues est égal à 1000 des simulations avec un résultat correct sur 1000 simulations (voir les Figures B.1(a) et B.1(b)).
  Par contre, sous l'hypothèse *MAR*, la méthode *NIPALS-PLS* pour le critère Q^2-*LOO* n'est meilleure que lorsque $n = 500$ et $p = 20$ (voir la Figure B.2(b)). Ses performances sont presque parfaites, dans ce cas, pour déterminer le nombre vrai de composantes.
  Cela signifie que si n est très grand par rapport à p, la méthode *NIPALS-PLS* avec le critère Q^2-*LOO* est la meilleure méthode pour déterminer le nombre vrai de composantes soit sous l'hypothèse *MCAR* soit sous l'hypothèse *MAR*.

- **Q^2-10-Fold**
  Les performances de la méthode *KNNimpute* avec le critère Q^2-10-*Fold* sont généralement les meilleures, que ce soit sous l'hypothèse *MCAR* ou sous l'hypothèse *MAR* (voir les Figures B.3 et B.4). Cette méthode choisit t^* presque parfaitement dans plus de 95,8% des simulations (sous l'hypothèse *MCAR*) et 94,3% des simulations (sous l'hypothèse *MAR*) quelle que soit la proportion de données manquantes sauf lorsque $d > 35\%$ dans le cas $n = 60$.

La deuxième meilleure méthode est la méthode *MICE* lorsque n est supérieur à p. Par exemple dans nos travaux est lorsque $n = 100$ et $n = 500$.

En revanche, les performances de la méthode *NIPALS-PLS* sur ce critère sont les pires. Presque toutes ses performances sont proches de zéro simulation avec un choix correct du nombre de composantes.

- **AIC**
  Les performances de ce critère sont très mauvaises, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*, et ceci avec n'importe quelle méthode (Figures B.5 et B.6). Dans la plupart des cas, le nombre de fois que le bon nombre de composantes est sélectionné par ce critère est proche de zéro. Il a tendance à sélectionner systématiquement plus de composantes que le nombre vrai de composantes.

- **AIC-DoF**
  Les performances d'*AIC-DoF* sont globalement très faibles pour déterminer t^* quelle que soit la méthode et quelle que soit l'hypothèse utilisée (Figures B.7 et B.8).

- **BIC**
  Ce critère fournit une performance presque identique à la performance de l'*AIC*. Presque toutes les performances des méthodes sont proches de zéro simulation avec un choix correct du nombre de composantes aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR* (Figures B.9 et B.10).

- **BIC-DoF**
  Les performances de les méthodes *KNNimpute* et *MICE* sur $n = 500$ et $n = 100$ sont assez bonnes lorsque la proportion de données manquantes est petite, bien qu'il existe des motifs de performance variables en fonction des dimensions.
  Le premier motif apparaît lorsque $n = 500$. Les performances dans ce cas diminuent avec l'augmentation de la proportion de données manquantes (voir les Figures B.11 et B.12 sur (a) et (b)).
  Le deuxième motif, qui apparaît dans le cas $n = 100$, est irrégulier avec des performances qui diminuent et puis augmentent avec la proportion de données manquantes (voir les Figures B.11 et B.12 sur (c)).
  Le troisième motif, propre aux cas $n = 80$ et $n = 60$, est aussi irréguliers mais avec des performances en bosse (voir les Figures B.11 et B.12 sur (d) et (e)).

Nombre vrai de composantes égal à 4

- **Q^2-LOO**
  En général, les performances de la méthode *MICE* avec ce critère donnent les meilleurs résultats aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR* (Figures B.13 et B.14). Ses performances diminuent avec l'augmentation de la proportion de données manquantes sauf pour les dimensions $n >> p$.

Des résultats intéressants ont également été obtenus dans le cas de $n >> p$. Les performances de trois méthodes : *NIPALS-PLS* (par *standard* ou *adaptative*), *MICE* et *KNNimpute* sont presque parfaites quelle que soit la proportion de données manquantes, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*. Ils choisissent t^* presque parfaitement sur 1000 simulations (voir les Figures B.13 et B.14 sur (a) et (b)).

- **Q^2-10-Fold**
  Les performances de la méthode *MICE* avec ce critère sont également les meilleures, comme dans le critère cas du Q^2-*LOO* précédent, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR* (Figures B.15 et B.16). Ses performances diminuent avec l'augmentation de proportion de données manquantes sauf pour les dimensions $n >> p$.
  Avec ce critère, les performances des méthodes *MICE* et *KNNimpute* sont presque parfaites lorsque $n >> p$ et quelle que soit la proportion de données manquantes aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR* (voir les Figures B.15 et B.16 sur (a) et (b)).

  En revanche, les performances de la méthode *NIPALS-PLS* sont les pires pour toutes les dimensions aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*. Presque toutes ses performances sont proches de zéro simulation avec un choix correct du nombre de composantes.

- **AIC**
  Les performances de toutes les méthodes de ce critère sont mauvaises aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*, et ceci avec n'importe quelle dimension de données (voir les Figures B.17 et B.18). Dans la plupart des cas, le nombre de fois que le bon nombre de composantes est sélectionné de ce critère est proche de zéro. Ce critère tende à sélectionner plus de composantes que les vraies composantes.

- **AIC-DoF**
  Les performances de la méthode *MICE* de ce critère montrent de bonnes performances en ayant deux modèles différents (voir les Figures B.19 et B.20).
  Le premier modèle au cas $n = 500$ est un modèle stable. Ce modèle se produit les mêmes performances pour quelle que soit la proportion de données manquantes (voir les Figures B.19 et B.20) sur (a) et (b)). Cette méthode choisit t^* à plus de 76,4% des simulations (l'hypothèse *MCAR*) et dans 77,1% des simulations (l'hypothèse *MAR*).
  Le deuxième modèle au cas $n = 100$ et $n = 80$ est irrégulier. Le nombre de composantes retenues diminue puis augmente avec l'augmentation de la proportion de données manquantes et ses performances sont assez bonnes lorsque la proportion de données manquantes est petite ($d < 15\%$).

  De plus, les performances de la méthode *MICE* ne sont pas bonnes lorsque la dimension $n = 60$, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*.

- **BIC**
  Les performances de ce critère sont les mêmes que les performances d'*AIC*. Presque

toutes les performances sont proches de zéro simulation avec un choix correct du nombre de composantes, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR* (voir les Figures B.21 et B.22) .

- **BIC-DoF**
  Les performances de ce critère sont presque les mêmes performances que le critère *AIC-DoF* aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*, mais ils choisissent plus le nombre vrai de composantes que l'*AIC-DoF* (voir les Figures B.23 et B.24). Il y a également deux modèles différents : le modèle stable au cas $n = 500$ et le modèle irrégulier au cas $n = 100$, 80 et 60.

  Les performances de la méthode *MICE* dans le cas $n >> p$ ($n = 500$) sont les meilleures quelle que soit la proportion de données manquantes et sous les deux hypothèses de données manquantes. Ils choisissent plus le nombre de vraies composantes, en moyenne de 86,4% des simulations (l'hypothèse *MCAR*) et 87,1% des simulations (l'hypothèse *MAR*).

  De plus, la méthode *KNNimpute* avec ce critère est également la meilleure performance lorsque $n = 500$ et $p = 100$, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR* (voir les Figures B.23 et B.24 sur (a)). Ses performances sont presque identiques comme la méthode *MICE* sur l'hypothèse *MCAR* mais non sur l'hypothèse *MAR*.

  Les performances de la méthode *SVDimpute* sont les pires. Presque toutes les performances de cette méthode sont proches de zéro simulation avec un choix correct du nombre de composantes, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*.

Nombre vrai de composantes égal à 6

- **Q^2-LOO**
  La méthode *MICE* avec ce critère est généralement la meilleure pour toutes les dimensions sauf à la dimension $n = 60$ sous les hypothèses *MCAR* et *MAR* (voir les Figures B.25 et B.26). Le nombre de composantes retenues de cette méthode diminue avec l'augmentation de la proportion de données manquantes sauf dans la dimension $n = 500$ et $p = 100$. Cette méthode donne plus de résultats sur le nombre de composantes lorsque la proportion de données manquantes est petite. Par exemple dans le cas $n = 80$, les performances sont bonnes quand $d < 25\%$ (l'hypothèse *MCAR*) et $d < 15\%$ (l'hypothèse *MAR*) (Figures B.25 et B.26 sur (d)).

  Au lieu de cela, nous voyons une grande différence dans les résultats de simulation sur la dimension $n = 500$ et $p = 100$ (voir les Figures B.25 et B.26 sur (a)). Les performances de toutes les méthodes sont les meilleures quelle que soit la proportion de données manquantes et quelle que soit l'hypothèse, même si le nombre de composantes retenues sous l'hypothèse *MCAR* est généralement plus proche du nombre vrai de composantes que sous l'hypothèse *MAR*.

  Le critère *Q^2-LOO* sur la dimension $n = 500$ et $p = 100$ choisit le nombre de composantes retenues presque parfaitement dans plus de 99,9% des simulations (la méthode *KNNimpute*), 99,9% des simulations (la méthode *MICE*), 94,2% des simulations (la méthode *NIPALS-PLS* avec *standard*), 95,5% des simulations (la méthode *NIPALS-PLS* avec *adaptative*) et 79,7% des simulations (la méthode *SVDimpute*) sur l'hypothèse

MCAR.

À l'hypothèse *MAR* sur la dimension $n = 500$ et $p = 100$, les performances de critère Q^2-*LOO* de toutes les méthodes sont également bonnes que l'hypothèse *MCAR* pour quelle que soit la proportion de données manquantes. Ils choisissent t^* dans plus de 99,9% des simulations (la méthode *MICE*), dans 99,1% des simulations (la méthode *KNNimpute*), dans 81,95% des simulations (la méthode *SVDimpute*), dans 80,9% des simulations (la méthode *NIPALS-PLS* avec *standard*) et dans 78,4% des simulations (la méthode *NIPALS-PLS* avec *adaptative*).

- **Q^2-10-Fold**
  Les performances de la méthode *MICE* avec ce critère sont les meilleures, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*, sauf à la dimension $n = 60$. Toutes les méthodes ne fonctionnent pas bien dans dimension $n = 60$ (Figures B.27 et B.26 sur (e)).

  Il existe trois conditions différentes sur les résultats de simulations obtenus.
  La première condition est sur les dimensions de données $n = 100$ et $n = 80$. La méthode *MICE* montre de bonne performance lorsque la proportion de données manquantes est petite (Par exemple voir les Figures B.27 et B.28 sur (c) et (d)). Le nombre de composantes retenues diminue avec l'augmentation de la proportion de données manquantes.

  La deuxième condition est sur les dimensions de données $n = 500$ et $p = 20$. Deux modelés existent dans cette condition.

  1. Le premier modèle est un modèle stable. Par exemple, les performances de la méthode *MICE* sont presque parfaites quelle que soit la proportion de données manquantes. Dans ce cas, le nombre de composantes retenues avec un résultat correct est égal à plus de 99,3% des simulations (l'hypothèse *MCAR*) et dans 98,5% des simulations (l'hypothèse *MAR*) (Figures B.27 et B.28 sur (b)).
  2. Le deuxième modèle est un modèle avec diminution. Par exemple, le nombre de composantes retenues par la méthode *KNNimpute* sur cette dimension est bon lorsque la proportion de données manquantes est petite ($d < 20\%$) à la fois sous l'hypothèse *MCAR* et sous l'hypothèse *MAR* avec un résultat correct. Leurs performances diminuent avec l'augmentation de la proportion de données manquantes.

  La troisième condition est sur les dimensions de données $n = 500$ et $p = 100$. Non seulement la méthode *MICE* fonctionne mieux comme les deux conditions ci-dessus, mais aussi les méthodes *KNNimpute* et *SVDimpute* donnent également de bons résultats sur les deux types hypothèses de données manquantes (*MCAR* et *MAR*), quelle que soit la proportion de données manquantes (voir les Figures B.27 et B.28 sur (a)).

  En revanche, les performances de la méthode *NIPALS-PLS* sont les pires. Presque toutes les performances *NIPALS-PLS* (le nombre de fois que le bon nombre de composantes est sélectionné) sont proches de zéro simulation.

- **AIC**
  Toutes les méthodes de ce critère n'obtiennent pas de bonnes performances sous les

hypothèses *MCAR* et *MAR* (Figures B.29 et B.30). Dans la plupart des cas, le nombre de fois que le bon nombre de composantes est sélectionné de ce critère est proche de zéro.

- **AIC-DoF**
  Dans la plupart des méthodes, les performances d'*AIC-DoF* sont faibles pour déterminer t^* sous les hypothèse *MCAR* et *MAR* (voir dans les Figures B.31 et B.32). Par exemple, le nombre de composantes retenues dans la méthode *MICE* ne sélectionne que la moitié des simulations pour choisir le nombre vrai de composantes.

- **BIC**
  Presque toutes les performances de ce critère, pour toutes les méthodes et toutes les dimensions, sont proches de zéro simulation correctes, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR* (Figures B.34 et B.34) .

- **BIC-DoF**
  Toutes les méthodes de ce critère donnent de moins performances aussi bien l'hypothèse *MCAR* que sous l'hypothèse *MAR* sauf la dimension $n = 500$ et $n = 100$. Les performances des méthodes *KNNimpute* et *MICE* dans la dimension $n = 500$ et $n = 100$ donnent d'assez bonnes performances (voir les Figures B.35 et B.36). Ils choisissent le nombre vrai de composantes en moyenne sous l'hypothèse *MCAR* : 75,07% des simulations (la méthode *MICE*) et 86,69% des simulations (la méthode *KNNimpute*) et sous l'hypothèse *MAR* : 72,9% des simulations (la méthode *MICE*) et 85,7% des simulations (la méthode *KNNimpute*).

7.1.2 Tableaux de données de forme horizontale

Les Figures dans l'Annexe B.2 présentent la fréquence du nombre de composantes retenues sur les données à matrice horizontale ($n = 20$ et $n = 40$) en fonction de la proportion de données manquantes (de 5% à 50% par 5%). Les résultats sont exprimés en nombre de fois, sur 1000 simulations et sous l'hypothèse *MCAR* ou sous l'hypothèse *MAR*, que le nombre de composantes retenues est égal au nombre vrai de composantes (t^*). Les performances de chaque critère pour une nombre variable de composantes à déterminer sont détaillées dans la suite.

Nombre vrai de composantes égal à 2

- **Q^2-LOO**
  Les performances de la méthode *NIPALS-PLS* sous l'hypothèse *MCAR*, qu'elle soit *standard* ou *adaptative*, donnent d'assez bons résultats du nombre de composantes retenues, lorsque la proportion de données manquantes est petite. Par exemple sur la dimension $n = 40$, ses performances sont bonnes lorsque $d < 35\%$ et sur la dimension $n = 20$ lorsque $d < 15\%$. De plus, les performances des méthodes *KNNimpute* et *SVDimpute* sont également proches des performances de la méthode *NIPALS-PLS* lorsque $n = 40$ et la

proportion de données manquantes est petite (Figure B.37).

En revanche sous l'hypothèse *MAR*, les performances de la méthode *NIPALS-PLS* avec ce critère, soit *standard* ou soit *adaptative*, sont très faibles (Figure B.38). Dans plus, les performances des méthodes *KNNimpute* et *SVDimpute* sont d'assez bonnes performances, sous $n = 40$ et $d < 35\%$, en moyenne dans 67,8% des simulations (*KNNimpute*) et dans 64,8% des simulations (*SVDimpute*).

- **Q^2-10-Fold**
  Les performances des méthodes *KNNimpute* et *SVDimpute* avec ce critère sont les meilleures soit sous l'hypothèse *MCAR* soit sous l'hypothèse *MAR* (Figures B.39 et B.40). En général, le nombre de composantes retenues augmente avec la diminution de la proportion de données manquantes.

  En revanche, les performances de la méthode *NIPALS-PLS* avec ce critère sont mauvaises. Presque toutes ses performances sont proches de zéro simulation avec un choix correct du nombre de composantes.

- **AIC**
  Les performances de toutes les méthodes de ce critère sont mauvaises soit sous l'hypothèse *MCAR* soit l'hypothèse *MAR*. Le nombre de fois que le bon nombre de composantes est sélectionné de ce critère est généralement proche de zéro avec un choix correct du nombre de composantes (voir les Figures B.41 et B.42).

- **AIC-DoF**
  Les performances d'*AIC-DoF* sont très faibles et irrégulières sur les deux types d'hypothèses de données manquantes (Figures B.43 et B.44).

- **BIC**
  Ce critère donne une performance qui est presque identique aux performances d'*AIC*. Presque toutes les performances de toutes les méthodes sur les deux types d'hypothèses de données manquantes sont proches de zéro simulation avec un choix correct du nombre de composantes (Figures B.45 et B.46).

- **BIC-DoF**
  Ce critère a presque les mêmes performances que les performances d'*AIC-DoF* pour toutes les méthodes et les deux types d'hypothèses de données manquantes. Leurs performances diminuent et augmentent irrégulièrement avec l'augmentation de la proportion de données manquantes (Figures B.47 et B.48).

Nombre vrai de composantes égal à 4
Presque tous les critères et toutes les méthodes montrent de mauvaises performances soit sous l'hypothèse *MCAR* ou sous l'hypothèse *MAR* (les Figures dans l'Annexe B.2.2). La méthode *MICE* avec les critères Q^2-*LOO* et Q^2-10-*Fold* n'a que de bonnes performances lorsque $d = 5\%$

sur la dimension $n = 40$ soit l'hypothèse *MCAR* soit l'hypothèse *MAR* (voir les Figures B.49, B.50, B.51 et B.52 sur (a).

Nombre vrai de composantes égal à 6
Toutes les critères et toutes les méthodes n'obtiennent pas de bonnes performances soit l'hypothèse *MCAR* soit l'hypothèse *MAR* (les Figures dans l'Annexe B.2.3).

7.2 Comparaison des performances en fonction des mécanismes et de la proportion de données manquantes

Les hypothèses de données manquantes : *MCAR* et *MAR*
Les performances des critères et des méthodes en fonction de la proportion de données manquantes sous l'hypothèse *MCAR* sont globalement meilleures que sous l'hypothèse *MAR* pour toutes les dimensions testées de jeux de données. Le nombre de composantes retenues sous l'hypothèse *MCAR* est plus proche du nombre vrai de composantes que l'hypothèse *MAR*.

Par exemple dans le cas $t^* = 2$, le critére Q^2-*LOO*, $n = 60$ et $d = 5\%$, la méthode *NIPALS-PLS* avec *standard* sous l'hypothèse *MCAR* a tendance à choisir plus de vraies composantes que sous l'hypothèse *MAR*. Il sélectionne le bon nombre de vraies composantes dans 99,5% des cas, mais seulement dans 58,5% des cas sous l'hypothèse *MAR* (voir les Figures B.1(e) et B.2(e) dans l'Annexe B.1).

La proportion de données manquantes
La proportion de données manquantes affecte les performances de tous les critères et de toutes les méthodes sur la détermination du nombre de composantes de la régression *NIPALS-PLS*. En résumé, nous pouvons dire qu'il existe trois modèles de performances différents :

- **Le modèle stable**. Ce modèle se produit lorsque le nombre d'observations est bien supérieur au nombre de variables. Nous notons les mêmes performances pour quelle que soit la proportion de données manquantes. Par exemple, la méthode *NIPALS-PLS* avec le critère Q^2-*LOO* sous l'hypothèse *MCAR* dans le cas $t^* = 2$ et $n = 500$ (voir les Figures B.1 (a) et (b)).

- **Le modèle avec diminution**. Les performances de ce modèle dépendent de la proportion de données manquantes. Le nombre de composantes retenues diminue avec l'augmentation de la proportion de données manquantes. Par exemple, la méthode *NIPALS-PLS* avec le critère Q^2-*LOO* sous l'hypothèse *MCAR* dans le cas $t^* = 4$ et $n = 100$ (voir les Figures B.13 (c)).

- **Le modèle irrégulier**. Les performances de ce modèle ne dépendent pas de la proportion de données manquantes. Le nombre de composantes retenues diminue puis augmente avec l'augmentation de la proportion de données manquantes. Par exemple, la méthode *MICE* avec le critère *AIC-DoF* sous l'hypothèse *MCAR* dans le cas $t^* = 4$ et $n = 100$ et $n = 80$ (voir les Figures B.19 (c) et (d)).

7.3 Comparaison des différents temps de calculs

Le Tableau 7.1 montre les moyennes de temps d'exécution d'une simulation (en secondes) calculée sur 1000 simulations sur des données incomplètes, sous les hypothèses *MCAR* et *MAR*. Le temps de calcul dépend généralement des dimensions du jeu de données, du type de données manquantes, des proportions de données manquantes et des méthodes de sélection utilisées.

Chaque dimension du jeu de données et chaque méthode a un effet différent sur le temps de calcul. Par exemple, la moyenne de temps d'exécution de la méthode *SVDimpute* sur la dimension n = 500 et p = 100 nécessite le temps le plus long pour une simulation. Elle était environ 5 fois plus longue que pour la méthode *NIPALS-PLS* lorsque t^* = 4 sous l'hypothèse *MCAR* (voir colonne 3 et lignes 1 et 4 du Tableau 7.1 (b) sous l'hypothèse *MCAR*). En revanche, lorsque t^* = 4, sous l'hypothèse *MCAR*, n = 500 et p = 20, la méthode *NIPALS-PLS* nécessite le temps le plus long pour une simulation par rapport à la *SVDimpute*. Elle était environ 5 fois plus longue que la méthode *SVDimpute* (voir colonne 4 et lignes 1 et 4 du Tableau 7.1 (b) sous l'hypothèse *MCAR*).

Le temps d'exécution sous l'hypothèse *MCAR* était plus long que sous l'hypothèse *MAR* dans la matrice verticale. En revanche, le temps d'exécution sous l'hypothèse *MAR* était plus long que sous l'hypothèse *MCAR* dans la matrice horizontale. Nous présentons également les résumés du temps de calculs qui sont les plus courts et sont les plus longs pour chaque composante à l'Annexe C.

7.4 Conclusion de simulations en données incomplètes

La détermination du nombre de composantes à inclure dans une régression *PLS* en données incomplètes a été faite en combinaison de critères et de méthodes avec des proportions différentes de données manquantes (de 5% à 50%) sous différentes hypothèses de type de données manquantes (*MCAR* et *MAR*). Les résultats des simulations mettent en évidence les points suivants.

1. Les performances des critères Q^2-*LOO* et Q^2-10-*Fold* dans la détermination du nombre vrai de composantes augmente généralement à mesure que le nombre d'observations augmente et, comme prévu, diminue lorsque la proportion de données manquantes augmente, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*.

2. Les performances des critères *AIC* et *BIC* sont très mauvaises, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*. Dans la plupart des cas, les nombres de composantes qui sont obtenus avec ces critères ont tendance à augmenter, c'est-à-dire à sélectionner plus de composantes qu'il n'en faut.

3. Le nombre de composantes retenues sous l'hypothèse *MCAR* est généralement plus proche du nombre vrai de composantes que sous l'hypothèse *MAR*.

4. Lorsque le nombre d'observations est beaucoup plus grand que le nombre de variables, par exemple n = 500 dans notre travail, la méthode *MICE* avec les critères Q^2-*LOO* et Q^2-10-*Fold* pour choisir t^* est généralement la meilleure, quelle que soit la proportion de données manquantes et quel que soit t^* sous les hypothèses *MCAR* ou *MAR*.

Table 7.1 – Moyenne de temps d'exécution d'une simulation (en secondes) calculée à partir de la moyenne de la proportion de données manquantes sur 1000 simulations.

(a) Nombre vrai de composantes égal à 2

Hypothèses	Méthodes	Dimension de données						
		$n = 500$ et $p = 100$	$n = 500$ et $p = 20$	$n = 100$ et $p = 20$	$n = 80$ et $p = 25$	$n = 60$ et $p = 33$	$n = 50$ et $p = 40$	$n = 20$ et $p = 100$
MCAR	*NIPALS-PLS*	6,9016	19,2454	10,6334	6,5865	2,9004	5,8008	2,0690
	MICE	24,7413	8,5215	28,5499	8,4735	15,8473	7,2825	3,1331
	KNNimpute	25,8378	4,3192	8,9867	7,8198	5,2882	8,7572	13,4556
	SVDimpute	30,5257	4,0798	5,7530	5,2656	5,4061	4,0774	1,8552
MAR	*NIPALS-PLS*	6,6281	16,7368	5,6074	4,9888	3,1581	2,5430	2,0135
	MICE	24,6380	7,6560	8,3539	6,5958	11,2006	9,0749	6,6804
	KNNimpute	25,7722	3,8853	7,1767	8,1590	8,6668	11,2182	12,8193
	SVDimpute	30,4676	3,6704	4,8971	5,5458	5,3505	4,9178	2,2685

(b) Nombre vrai de composantes égal à 4

Hypothèses	Méthodes	Dimension de données						
		$n = 500$ et $p = 100$	$n = 500$ et $p = 20$	$n = 100$ et $p = 20$	$n = 80$ et $p = 25$	$n = 60$ et $p = 33$	$n = 50$ et $p = 40$	$n = 20$ et $p = 100$
MCAR	*NIPALS-PLS*	4,8570	18,1571	1,1223	1,0502	0,8922	0,8403	0,9730
	MICE	18,4087	8,5171	26,6120	32,9364	18,2981	10,3448	4,7432
	KNNimpute	16,8760	4,1800	0,9890	1,1801	1,5319	2,3652	6,8520
	SVDimpute	22,3557	3,9893	0,7546	0,8978	1,1006	1,2574	0,7929
MAR	*NIPALS-PLS*	5,7105	17,7152	0,9088	0,7566	0,6174	0,6220	1,0089
	MICE	17,5213	8,8006	30,8164	34,8002	17,6483	11,2101	10,7938
	KNNimpute	16,5886	4,2644	1,1477	1,2234	1,5162	2,4108	7,4192
	SVDimpute	22,2568	4,0245	0,8393	0,9169	1,1020	1,2523	0,8668

(c) Nombre vrai de composantes égal à 6

Hypothèses	Méthodes	Dimension de données						
		$n = 500$ et $p = 100$	$n = 500$ et $p = 20$	$n = 100$ et $p = 20$	$n = 80$ et $p = 25$	$n = 60$ et $p = 33$	$n = 50$ et $p = 40$	$n = 20$ et $p = 100$
MCAR	*NIPALS-PLS*	6,6250	17,9773	8,8798	7,9510	5,8231	3,8536	2,6180
	MICE	25,1497	8,8154	6,1144	4,0283	7,7597	6,2529	5,0204
	KNNimpute	25,7668	4,1400	8,4355	9,4448	11,1870	14,0009	15,5334
	SVDimpute	30,4574	4,0033	5,5786	6,1963	6,5918	5,6169	2,1297
MAR	*NIPALS-PLS*	6,9194	17,1589	6,4234	5,6983	4,1767	2,5925	2,4249
	MICE	18,3441	8,8070	6,5518	4,4048	7,0691	7,8233	9,2974
	KNNimpute	25,9299	4,1096	8,2998	9,3873	10,4195	12,6226	9,5938
	SVDimpute	30,6005	3,9768	5,4988	6,1302	6,3525	5,1142	2,2425

Nous avons résumé les résultats des simulations de toutes les combinaisons des critères et des méthodes qui donnent une estimation du nombre de composantes plus proche du nombre vrai de composantes dans le Tableau 9.2 du Chapitre 9. Ce tableau ne montre que six tailles des jeux de données, la dimension $n = 500$ et $p = 100$ en étant exclue. Nous présentons cette dimension $n = 500$ et $p = 100$ au Tableau 7.2 en dessous. Les résultats de ce tableau sont exprimés en combinaison de critères et de méthodes avec une estimation du nombre de composantes plus proche du nombre vrai de composantes (t^*) sur 1000 simulations.

En conclusion, quel que soit le critère utilisé, le type de données manquantes et la proportion de données manquantes doivent également être pris en compte car ils influent sur le nombre de composantes sélectionnées.

Table 7.2 – Évaluation des méthodes *NIPALS-PLS*, *MICE*, *KNNimpute* et *SVDimpute* sur le cas $n = 500$ et $p = 100$.

t^*	Hypothèses	Critères					
		Q^2-*LOO*	Q^2-10-*Fold*	*AIC*	*AIC-DoF*	*BIC*	*BIC-DoF*
2	*MCAR*	*NIPALS – PLS* *KNNimpute* *MICE* *SVDimpute*	*KNNimpute* *MICE* *SVDimpute*	-	-	-	*KNNimpute* *MICE*
	MAR	*MICE* *KNNimpute* *NIPALS – PLS* *SVDimpute*	*MICE* *KNNimpute* *SVDimpute*	-	-	-	*KNNimpute* *MICE*
4	*MCAR*	*MICE* *NIPALS – PLS* *KNNimpute* *SVDimpute*	*MICE* *KNNimpute* *SVDimpute*	-	-	-	*KNNimpute* *MICE*
	MAR	*MICE* *NIPALS – PLS* *KNNimpute* *SVDimpute*	*MICE* *KNNimpute* *SVDimpute*	-	*MICE*	-	*MICE* *KNNimpute*
6	*MCAR*	*KNNimpute* *MICE* *NIPALS – PLS* *SVDimpute*	*MICE* *KNNimpute* *SVDimpute*	-	-	-	*KNNimpute* *MICE*
	MAR	*MICE* *KNNimpute* *NIPALS – PLS* *SVDimpute*	*MICE* *KNNimpute* *SVDimpute*	-	-	-	*MICE* *KNNimpute*

Chapitre 8

Pré-traitement de données réelles

Dans ce travail, nous avons quatre jeux de données réelles avec une réponse $\mathbf{y}$ univariée. Ces quatre jeux de données réelles sont divisés en deux types selon les dimensions de la matrice des prédicteurs $\mathbf{X}$: matrice verticale ($n > p$) ou matrice horizontale ($n < p$). Nous analysons et comparons ensuite les résultats des simulations et les résultats sur les données réelles expliquant les effets des données manquantes sur la détermination du nombre de composantes de la régression *NIPALS-PLS*.

Dans ce chapitre, nous expliquerons brièvement la structure de colinéarité des données réelles. Pour mesurer la colinéarité, l'extension du *package* `mctest` dans le logiciel R peut être utilisée (Ullah and Aslam, 2018). L'approche la plus classique consiste à utiliser les FIV. La valeur des FIV estime dans quelle mesure la variance d'un coefficient est «augmentée» en raison d'une relation linéaire avec les autres variables explicatives.

8.1 Données à matrice verticale

Pollution à l'ozone à Los Angeles
Le jeu de données de la pollution à l'ozone à Los Angeles en 1976 peut être téléchargé en ligne dans le *package* : `mlbench` dans le logiciel R (Leisch and Dimitriadou, 2010). Ce jeu de données concerne les 12 variables explicatives contenant la date de la mesure et des informations sur la vitesse du vent, l'humidité, la température, etc. L'ensemble des données a identifié 366 observations et chaque observation en une journée permet de prédire la moyenne quotidienne maximale de la pollution à l'ozone. Les données d'origine contiennent les données manquantes. Nous avons donc utilisé 203 observations sans données manquantes.

Le Tableau 8.1 montre les valeurs de la corrélation entre les variables explicatives et de la corrélation entre les variables explicatives et la variable réponse qui sont supérieure à 0.7. Les corrélations entre les variables explicatives $\mathbf{x}_4$, $\mathbf{x}_8$, $\mathbf{x}_9$, $\mathbf{x}_{10}$ et $\mathbf{x}_{12}$ sont élevées. De plus, les variables explicatives : $\mathbf{x}_8$, $\mathbf{x}_9$ et $\mathbf{x}_{12}$ sont également fortement corrélées avec la réponse $\mathbf{y}$.

Les diagnostics de colinéarité des FIV de ce jeu de données de la pollution à l'ozone à Los Angeles sont représentés à la Figure 8.1. Presque toutes les FIV sont supérieures à 1, cela indique

donc que les variables explicative sont corrélées. Ainsi, ils indiquent que la méthode des MCO de la régression linéaire multiple n'est pas appropriée en raison de cette colinéarité.

Table 8.1 – Valeurs des corrélations entre les variables du jeux de données sur la pollution à l'ozone à Los Angeles pour les corrélation de valeurs absolues supérieures à 0.7.

Variable	$\mathbf{x}_4$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{y}$
$\mathbf{x}_8$	0.772470				0.806330
$\mathbf{x}_9$	0.759022	0.913962			0.896893
$\mathbf{x}_{12}$	0.717566	0.843103	0.930810	0.782861	0.856421

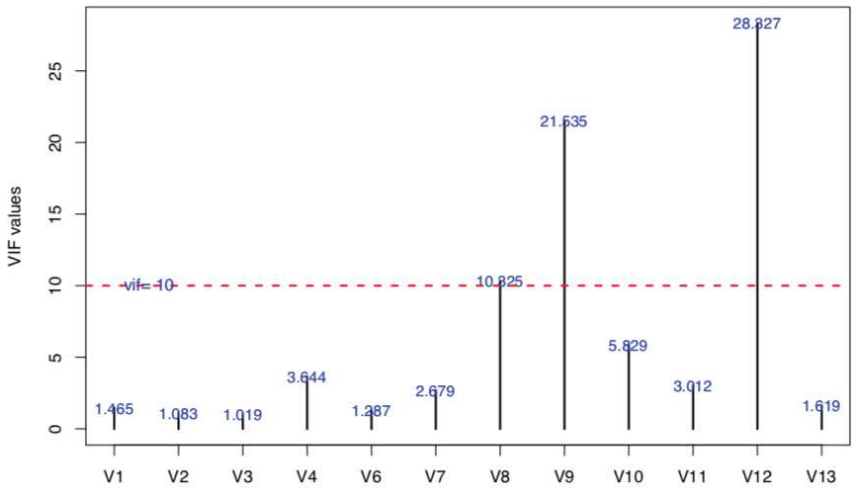


Figure 8.1 – Diagnostics de colinéarité du jeu de données de pollution à l'ozone à Los Angeles avec V = les variables explicatives.

Tétracycline

Cet exemple concerne le jeu de données de Goicoechea and Olivieri (1999b). Il s'agit d'une étude qui a examiné la possibilité de mesurer la concentration sérique de la tétracycline et de ses dérivés dans un échantillon de sang en appliquant des mesures synchrones de spectrofluorométrie. La tétracycline a été obtenue auprès des laboratoires Richet et sa pureté a été vérifiée conformément aux recommandations de la Pharmacopée. La tétracycline exerce diverses activités antimicrobiennes contre les bactéries à Gram positif et à Gram négatif.

Ce jeu de données contient 107 observations sur 101 variables explicatives qui sont des valeurs dans une gamme spectrale (nm) comprise entre 0 et 600. Les valeurs des FIV montrent des valeurs qui sont supérieures à 10, ce qui indique qu'il existe des multicolinéarités élevée dans le jeu de données. Cette situation indique que l'analyse de régression linéaire multiple ne peut pas être utilisée, ce qui constitue une des faiblesses de la régression avec la méthode des MCO : la limitation par les multicolinéarités entre variables explicatives.

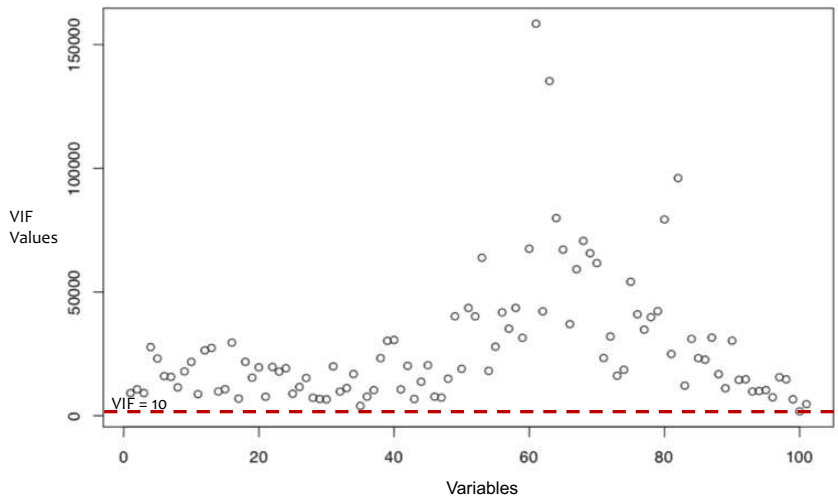


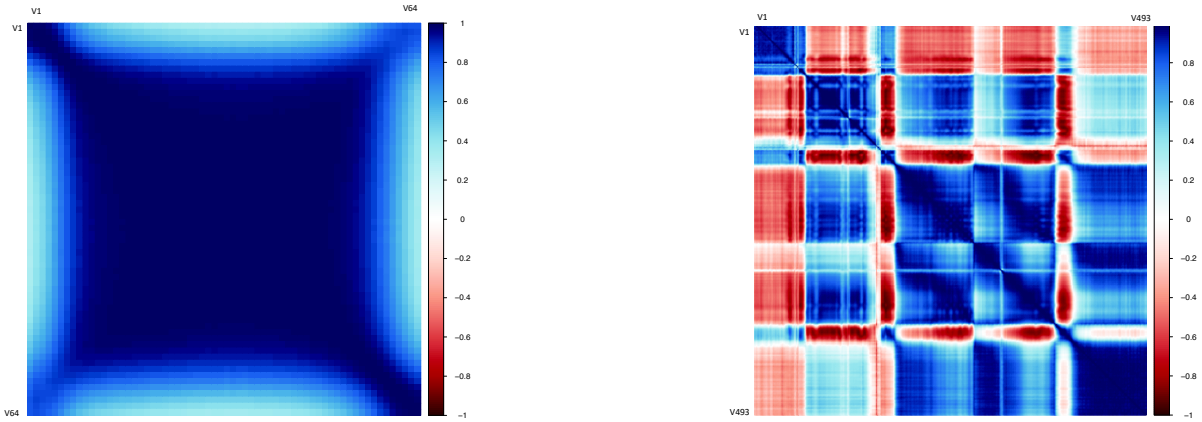
Figure 8.2 – Diagnostics de colinéarité du jeu de données de Tétracycline.

8.2 Données à matrice horizontale

Il y a deux exemples de données réelles dans ce cas. Le premier jeu de données réelles est celui des données de **Bromhexine** qui comporte 23 observations et 64 concentrations de Bromhexine dans un sirop pharmaceutique ($n = 23 \times p = 64$) (Goicoechea and Olivieri, 1999a). Le second jeu de données est des données d'**Octane** dans lequel Goicoechea and Olivieri (2003) a noté l'indice d'octane mesuré dans différents échantillons d'essences à partir de mesure faites dans le proche infrarouge (en anglais *near infrared (NIR)*). Les données portent sur 68 échantillons d'essence à spectre NIR avec 493 variables explicatives ($n = 68 \times p = 493$).

Avec plus de variables explicatives que le nombre d'observations, ces jeux de données ne peuvent pas utiliser la régression linéaire multiple. La régression *PLS* est donc l'une des solutions pour la modélisation de ce type de données.

Dans ces exemples de jeux de données, il existe de fortes corrélations entre plusieurs variables explicatives. Nous présentons les valeurs de corrélation des jeux de données de Bromhexines et d'Octane respectivement sur les Figures 8.3a et 8.3b.



(a) Bromhexine.

(b) Octane.

Figure 8.3 – Diagnostics de colinéarité.

Chapitre 9

Performance de différents algorithmes de régression NIPALS-PLS sur des données incomplètes

Ce chapitre consiste en un article publié en 2019 dans le journal *Statistical Applications in Genetics and Molecular Biology* avec le titre *Determining the Number of Components in PLS Regression on Incomplete Data Set*. L'article explique l'essentiel de nos travaux qui ont été menés sur les problématiques de détermination du nombre de composantes dans la régression *NIPALS-PLS* et son application aux données réelles.

Abstract

Partial least squares regression—or PLS regression—is a multivariate method in which the model parameters are estimated using either the SIMPLS or NIPALS algorithm. PLS regression has been extensively used in applied research because of its effectiveness in analyzing relationships between an outcome and one or several components. Note that the NIPALS algorithm can provide estimates parameters on incomplete data. The selection of the number of components used to build a representative model in PLS regression is a central issue. However, how to deal with missing data when using PLS regression remains a matter of debate. Several approaches have been proposed in the literature, including the Q^2 criterion, and the *AIC* and *BIC* criteria. Here we study the behavior of the NIPALS algorithm when used to fit a PLS regression for various proportions of missing data and different types of missingness. We compare criteria to select the number of components for a PLS regression on incomplete data set and on imputed data set using three imputation methods : multiple imputation by chained equations, *k*-nearest neighbour imputation, and singular value decomposition imputation. We tested various criteria with different proportions of missing data (ranging from 5% to 50%) under different missingness assumptions. Q^2-leave-one-out component selection methods gave more reliable results than *AIC* and *BIC*-based ones.

keywords : PLS regression, Number of Components, Missing Data, NIPALS, Imputation Method

9.1 Introduction

Missing data are present in many real-world data set and often cause problems in data analysis. Missing data can occur for many reasons, including uncollected data, mishandled samples, equipment errors, measurement errors, misunderstanding questionnaires, etc.(Grung and Manne, 1998; Folch-Fortuny *et al.*, 2016). According to Little and Rubin (2002), missing data can be divided into three categories, namely : Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR). If the probability that the data is known depends neither on the observed value nor on the missing values, the data are said to be MCAR. In the case of MAR, missingness depends only on the values of the observed data. Lastly, data are said to be MNAR if missingness depends both on the observed and missing data values.

Many methods have been proposed for imputing missing data. The simplest ones rely on single value imputation, e.g., the mean over the complete cases in the study sample—known as mean imputation (Troyanskaya *et al.*, 2001). More complex methods include regression-based imputation (Horton and Lipsitz, 2001), imputation based on Non-linear Iterative PArtial Least Squares (NIPALS) (Tenenhaus, 1998), multiple imputation (Rubin, 1987), *K*-Nearest Neighbours imputation (KNNimpute) (Dixon, 1979), Singular Value Decomposition-based imputation (SVDimpute) (Troyanskaya *et al.*, 2001), and so on.

Partial least squares (PLS) regression was introduced in the 1970s by Wold (1966). It has gone from being popular in chemometrics (see Wold *et al.*, 2001) to being commonly used in many research areas such as bioinformatics (Nguyen and Rocke, 2004), medicine (Yang *et al.*, 2017), social sciences (Sawatsky *et al.*, 2015), and spectroscopy (Oleszko *et al.*, 2017). PLS regression—in its classical form—is based on the NIPALS algorithm. The alternative estimation method for PLS regression is SIMPLS algorithm, for *Straightforward Implementation of a statistically inspired Modification to PLS* (see De Jong, 1993b). The former has been implemented in software such as SIMCA (Eriksson *et al.*, 2002) and more recently in the `plsRglm` package (Bertrand *et al.*, 2014).

The NIPALS algorithm was initially devised to carry out principal component analysis (PCA) on incomplete data set. It explains why its reliability under increasing proportions of missing data has been studied mainly in this setting (Nelson *et al.*, 1996; Grung and Manne, 1998; Arteaga and Ferrer, 2002). As in PCA, one of the justifications for using the NIPALS algorithm in PLS regression is that it enables models to be fitted on incomplete data set. This feature is long-known and frequently used as an argument to apply this algorithm preferentially. In this paper, we focus on univariate PLS, also known as PLS1, and the use of the NIPALS algorithm.

The goal of PLS regression is to predict a set of dependent variables from a set of independent variables. This prediction is obtained by extracting from the predictors a set of orthogonal factors called components that have the best predictive power. Determining the optimal number of components is thus a critical problem in PLS regression. Selecting a less-than-optimal number of components leads to a loss of information, whereas selecting a more-than-optimal number can lead to models with poor predictive ability (Wiklund *et al.*, 2007).

Several papers have studied ways to determine the number of components to retain in the final PLS regression (see for instance Lazraq *et al.*, 2003). In spectroscopy, for example, the

sample spectrum is the sum of the spectra of the constituents multiplied by their concentration in the sample. This interpretation makes sense with the data explained by several components in PLS regression. As also mentioned by Goicoechea and Olivieri (1999a), the analyse of interest is embedded in a complex mixture of several components. Another interesting application of the number of components is the use of microscopic concepts such as molecules and reactions in chemical and biological data which closely correspond with the use of a number of components. This has been discussed by Burnham *et al.* (1999), Burnham *et al.* (1996), and Kvalheim (1992).

Though it is now considered a benchmark for incomplete data set analysis, the reliability of the NIPALS algorithm when estimating PLS regression parameters on incomplete data sets has been studied very little, despite its importance. In the context of PLS regression, details on to missing data, such as how to estimate scores on incomplete data, and the impact of missing data on PLS prediction have been reported by Nelson *et al.* (1996), Rännar *et al.* (1995), and Serneels and Verdonck (2008). However, the sensitivity of the NIPALS algorithm to increasing missing data proportions does not seem to have been given much attention. Moreover, in the few papers that pertain to incomplete data issues, the reliability of the NIPALS algorithm under different missingness mechanisms described in Little and Rubin (1987) has been systematically ignored.

In summary, at least two things may affect parameter estimates for PLS regression : the proportion of missing data and the type of missingness. Both issues will be studied here.

Besides, we compare criteria for selecting the number of components in PLS regression. The most-used method for incomplete data sets is PLS regression with the NIPALS algorithm. Other methods for imputing data sets are multiple imputation by chained equations (MICE), K-Nearest Neighbours imputation, and Singular Value Decomposition imputation. The influence of the proportion of missing data and the type of missingness on the estimation of the number of components in a PLS regression is the primary purpose of the present study.

The paper is organized as follows. Section 9.2 describes the methods, presenting a brief description of PLS regression, cross-validation with missing values, and imputation methods. Section 9.3 describes the simulation study, and Section 9.4 gives the results of this study. Real data are presented in Section 9.5. We conclude with a general discussion in Section 9.6.

9.2 Partial least squares regression and related works

9.2.1 PLS regression

A complete description of PLS regression can be found in Wold (1966) and Höskuldsson (1988).

Suppose that $\mathbf{X}$ is an $n \times p$ data matrix of continuous p explanatory variables $\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_p$ where p can be greater than n (the number of observations) and $\mathbf{y}$ is an univariate response variable.

PLS1 replaces the OLS goal of finding β that maximizes squared correlation $\text{cor}(\mathbf{X}\beta, \mathbf{y})^2$

with an alternative goal of finding β with length $||\beta|| = 1$ maximizing covariance

$$\text{cov}(\mathbf{X}\beta, \mathbf{y})^2 = \text{cor}(\mathbf{X}\beta, \mathbf{y})^2 \times \text{var}(\mathbf{X}\beta)$$

which effectively penalizes directions of low variance.

The PLS1 regression model is :

$$\mathbf{y} = \mathbf{T}\mathbf{q} + \varepsilon \tag{9.1}$$

where $\mathbf{T}$ is the $(n \times H)$ matrix of the H extracted score vectors (with columns $\mathbf{t}_h = (t_{1h}, \ldots, t_{nh})'$, $h = 1, \ldots, H$. -$\mathbf{A}'$ means the transpose of the $\mathbf{A}$ matrix or vector-), $\mathbf{q}$ is a vector -of loadings- of length (n), and ε is the error vector.

As stated before, the PLS regression can be included in an objective function framework (Burnham *et al.*, 1996) and viewed as the solution to an iterated maximization problem. The first component is the solution to :

$$\max_{\mathbf{w}_1} \text{Cov}(\mathbf{X}\mathbf{w}_1, \mathbf{y})^2 = \max_{\mathbf{w}_1} \mathbf{w}_1'\mathbf{X}'\mathbf{y}\mathbf{y}'\mathbf{X}\mathbf{w}_1 \quad \text{subject to} \quad \mathbf{w}_1'\mathbf{w}_1 = 1. \tag{9.2}$$

and the maximum for (9.2) is obtained at $\mathbf{w}_1$, the largest eigenvector of the matrix $\mathbf{X}'\mathbf{y}\mathbf{y}'\mathbf{X}$. In order to obtain further weight vectors, the algorithm is repeated with deflated **X-** matrix and **y-**vector. The deflation process is defined for $i = 1, 2, \ldots, h-1$ as

$$\mathbf{X}_{i+1} = \left(\mathbf{I} - \frac{\mathbf{t}_i\mathbf{t}_i'}{\mathbf{t}_i'\mathbf{t}_i}\right)\mathbf{X}_i, \quad \mathbf{y}_{i+1} = \left(\mathbf{I} - \frac{\mathbf{t}_i\mathbf{t}_i'}{\mathbf{t}_i'\mathbf{t}_i}\right)\mathbf{y}_i. \tag{9.3}$$

where $\mathbf{X}_1 = \mathbf{X}$, $\mathbf{y}_1 = \mathbf{y}$ and $\mathbf{t}_i = \mathbf{X}\mathbf{w}_i$. Hence only the subspace in $\mathbf{X}$ that is orthogonal to the earlier linear combinations developed in the $\mathbf{X}$-space is used. The $\mathbf{y}$-space is projected onto the space orthogonal to the previous $\mathbf{X}$-components. Subsequent weights vectors $\mathbf{w}_i$ are chosen to satisfy (9.2) using deflated $\mathbf{X}_i$-matrices and $\mathbf{y}_i$-vectors in place of the original $\mathbf{X}$-matrix and $\mathbf{y}$-vector.

The objective function can be updated to include the iterative deflation steps as shown in Burnham *et al.* (1996) for both univariate and multivariate PLS :

$$\max_{\alpha_i, \beta_i} \left(\alpha_i'\mathbf{X}'\mathbf{y}\beta_i - \sum_{j=1}^{i-1} \frac{(\alpha_i'\mathbf{X}'\mathbf{X}\mathbf{r}_j)(\mathbf{r}_j'\mathbf{X}'\mathbf{y}\beta_i)}{\mathbf{r}_j'\mathbf{X}'\mathbf{X}\mathbf{r}_j} \right) \quad \text{subject to } \alpha_i'\alpha_i = 1, \beta_i'\beta_i = 1$$

where $\mathbf{X}\mathbf{r}_j$ is given by the Gram-Schmidt formula for determining an orthogonal basis for a set of vectors. As a consequence, $\mathbf{X}\mathbf{r}_j$ can be expressed as

$$\mathbf{X}\mathbf{r}_j = \mathbf{X}\alpha_j - \sum_{k=1}^{j-1} \mathbf{X}\mathbf{r}_k \frac{\alpha_j'\mathbf{X}'\mathbf{X}\mathbf{r}_k}{\mathbf{r}_k'\mathbf{X}'\mathbf{X}\mathbf{r}_k}$$

In addition, in our univariate response setting, we must have $\beta_i = \beta_i = 1$. This leads to a simplified problem :

$$\max_{\alpha_i} \left(\alpha_i'\mathbf{X}'\mathbf{y} - \sum_{j=1}^{i-1} \frac{(\alpha_i'\mathbf{X}'\mathbf{X}\mathbf{r}_j)(\mathbf{r}_j'\mathbf{X}'\mathbf{y})}{\mathbf{r}_j'\mathbf{X}'\mathbf{X}\mathbf{r}_j} \right) \quad \text{subject to } \alpha_i'\alpha_i = 1 \tag{9.4}$$

In its classical form and with complete data, the PLS model fitting process (Rosipal and Krämer,

2006) is based on the nonlinear iterative partial least squares (NIPALS) algorithm that finds a sequence of weights vector $(\mathbf{w}_i)_{1,\ldots,i}$ solution to (9.4).

As a result, this fitting process for the PLS regression finds iteratively the new variables $\mathbf{t}_h$. They are called components and are mutually orthogonal. The number of components H can be chosen from data through cross validation, a model selection criterion (see next Section 9.2.3). The $\mathbf{t}_h$ PLS components are linear combinations of the p variables of the matrix $\mathbf{X}$ with formulas :

$$\mathbf{T} = \mathbf{XW} \tag{9.5}$$

where $\mathbf{W}$ is a $p \times H$ matrix of weights. The columns of $\mathbf{W}$ are denoted as $\mathbf{w}_h = (w_{1h}, \ldots, w_{ph})'$, respectively, for $h = 1, \ldots, H$.

9.2.2 NIPALS-PLSR

In the chapter 6 of Tenenhaus book (Tenenhaus, 1998), the NIPALS algorithm divides the data set into a complete part and an incomplete part to build a matrix, called $\mathbf{X}_h$. The columns of the matrix $\mathbf{X}_h$ are noted $\mathbf{x}_{h1}, \ldots, \mathbf{x}_{hj}, \ldots, \mathbf{x}_{hp}$.

The PLS regression which has been estimated by NIPALS algorithm (NIPALS-PLSR) starts with (optionally) transformed, scaled, and centered $\mathbf{X}$ and $\mathbf{y}$. The $\mathbf{t}_h$ component is equal to $\mathbf{Xw}_h$, where the weight $\mathbf{w}_h$ (see below in the step 4) is constructed step by step. The steps of NIPALS-PLSR follows in Algorithm 1.

Algorithm 1 NIPALS-PLSR algorithm in complete data set (Tenenhaus, 1998)

1: Initialize $\mathbf{X}_0 = \mathbf{X}, \mathbf{y}_0 = \mathbf{y}$
2: **for** $h = 1, 2, \ldots, H$ **do**
3: **repeat**
4: $\mathbf{w}_h = \mathbf{X}'_{h-1}\mathbf{y}_{h-1} / \mathbf{y}'_{h-1}\mathbf{y}_{h-1}$
5: Normalize $\mathbf{w}_h$ to 1
6: $\mathbf{t}_h = \mathbf{X}_{h-1}\mathbf{w}_h / \mathbf{w}'_h\mathbf{w}_h$
7: $\mathbf{p}_h = \mathbf{X}'_{h-1}\mathbf{t}_h / \mathbf{t}'_h\mathbf{t}_h$
8: $\mathbf{X}_h = \mathbf{X}_{h-1} - \mathbf{t}_h\mathbf{p}'_h$
9: $\mathbf{q}_h = \mathbf{X}'_{h-1}\mathbf{t}_h / \mathbf{t}'_h\mathbf{t}_h$
10: $\mathbf{y}_h = \mathbf{y}_{h-1} - \mathbf{t}_h q_h$
11: **until** Convergence of $\mathbf{p}_h$
12: **end for**

The NIPALS-PLSR algorithm can, therefore, be seen as a compromise between a multiple linear regression and a principal component analysis (Wold *et al.*, 1987), in which the first h components $\mathbf{t}_h$ are the principal components whose covariances with $\mathbf{y}$ are the largest.

There is an additional interest to use the NIPALS algorithm in PLS regression in the presence of incomplete data. Treatment of missing data with NIPALS-PLSR can be implicitly associated with a simple imputation method (Bastien and Tenenhaus, 2003). When data in any column or row of the matrix $\mathbf{X}$ is missing, the iterative regressions are performed using the available values, ignoring the missing ones (Tenenhaus, 1998).

The iteration of the NIPALS-PLSR for the h^{th} component follows in Algorithm 2.

Algorithm 2 NIPALS-PLSR algorithm in incomplete data set

Initialize $\mathbf{X}_1 = \mathbf{X}, \mathbf{y}_1 = \mathbf{y}$
for $h = 1, 2, \ldots, H$ **do**
  repeat
    $w_{j,h} = \frac{\sum_{\{i:x_{ij,h} \text{ and } y_{i,h} exist\}} x_{ij,h} y_{i,h}}{\sum_{\{i:x_{ij,h} \text{ and } y_{i,h} exist\}} y_{i,h}^2}$, $j = 1, \ldots, p$
    Normalize $\mathbf{w}_h$ to 1
    $t_{i,h} = \frac{\sum_{\{j:x_{ij,h} \ exists\}} x_{ij,h} w_{j,h}}{\sum_{\{j:x_{ij,h} \ exists\}} w_{j,h}^2}$, $i = 1, \ldots, n$
    $p_{j,h} = \frac{\sum_{\{j:x_{ij,h} \ exists\}} x_{ij,h} t_{i,h}}{\sum_{\{j:x_{ij,h} \ exists\}} t_{i,h}^2}$, $j = 1, \ldots, p$
    $\mathbf{X}_{h+1} = \mathbf{X}_h - \mathbf{t}_h \mathbf{p}_h'$ for $x_{ij,h}$ existing
    $q_h = \frac{\sum_{\{i:y_{i,h} \ exists\}} y_{i,h} t_{i,h}}{\sum_{\{i:y_{i,h} \ exists\}} t_{i,h}^2}$
    $\mathbf{y}_{h+1} = \mathbf{y}_h - \mathbf{t}_h q_h$ for $y_{i,h}$ existing
  until Convergence of $\mathbf{p}_h$
end for

9.2.3 Model selection : cross-validation and information criteria

Several papers have studied methods to determine the number of components to retain in the final model of PLS regression—see, for instance, Lazraq *et al.* (2003). In this present study, only selection of the number of components is considered, including all $\mathbf{x}_j$ variables on each of the first components, whatever their significance for these components. There exist several approaches in the literature to choose the h number of components to include in the final model : the Q^2 criterion, computed by cross-validation (Stone, 1974) and information criteria like the Akaike Information Criterion (*AIC*) (Akaike, 1969) and the Bayesian Information Criterion (*BIC*) (Schwartz, 1978). We are going to describe them briefly below.

Cross-validation is a practical and reliable way to test this predictive significance (Tenenhaus, 1998; Wakeling and Morris, 1993). It has become the standard in PLS regression analysis and incorporated in one form or another in all available PLS regression software. Cross-validation is performed by dividing the complete data set in k complete subsets, and then developing several parallel models from reduced data with one of the groups deleted. It is called k-Fold cross-validation. Five- to ten-Fold cross-validation are common. If $k = n$, this approach is called leave-one-out (LOO) cross-validation.

If we study an incomplete data set, so cross-validation requires modifications. There are two methods : *standard* cross-validation or *adaptive* cross-validation. The first, called *standard* cross-validation, is to predict the response value of any row of the data set as if it featured missing data. In *adaptive* cross-validation, it predicts the response value for a row accordingly to the presence of missing data in that row : regular prediction for complete data if no missing data in the row, missing data specific prediction if there are missing data in the row (Bertrand *et al.*, 2014).

The Q_h^2 criterion is defined for each h component as :

$$Q_h^2 = 1 - PRESS_h / RSS_{h-1}, \tag{9.6}$$

where $PRESS_h$ is the *PRedictive Error Sum of Squares* when the number of components containing h components and RSS_{h-1} is the *Residual Sum of Squares* associated to the model containing $(h-1)$ components. RSS_h can be calculated by $RSS_h = \sum_{i=1}^{n} \left(y_i - \hat{y}_{h,i}\right)^2$ where $\hat{y}_{h,i}$ is predicted value based on the model with h components.

PRESS is computed by cross validation with the below formula (Pérez-Enciso and Tenenhaus, 2003) :

$$PRESS_h = \sum_{i=1}^{n} \left(y_{h-1,i} - \hat{y}_{h-1,-i}\right)^2, \tag{9.7}$$

where $y_{h-1,i}$ is the residual of observation i when $h-1$ components are fitted, and $\hat{y}_{h-1,-i}$ is the predicted y_i obtained when the $i-th$ observation is removed.

A new $\mathbf{t}_h$ component is kept for the prediction of $\mathbf{y}$ if (see Tenenhaus, 1998) :

$$\sqrt{PRESS_h} \leq 0.95\sqrt{RSS_{h-1}} \Leftrightarrow Q_h^2 \geq 0.0975, \tag{9.8}$$

where (0.95) is arbitrary value.

For components selection by information criteria, the number of degrees of freedom (DoF) has been computed using the methods of Krämer and Sugiyama (2011) and implemented in the `plsdof` package (Krämer and Braun, 2019). The PLS routines in the `plsRglm` package are based on these DoF except in the case of incomplete data for which only naive DoF are currently implemented.

Krämer and Braun (2019) defined *AIC* and *BIC* as :

$$AIC = \frac{RSS}{n} + 2\frac{DoF}{n}\sigma^2 \tag{9.9}$$

$$BIC = \frac{RSS}{n} + log(n)\frac{DoF}{n}\sigma^2 \tag{9.10}$$

where RSS represents the Residual Sum of Squares as associated to the PLS regression, n is the number of observations, σ^2 is the unknown variance of the error variables and DoF is the degrees of freedom for the PLS regression (Krämer and Sugiyama, 2011).

9.2.4 Imputation methods

9.2.4.1 Multiple imputation

Multiple imputation is a general statistical method for the analysis of incomplete data sets (Rubin, 1987; Royston, 2004; Van Buuren, 2012). This method has become a conventional approach for dealing with missing data in numerous analyses from different domains. Multiple

imputation aims to provide unbiased and valid estimates of associations based on information from the available data.

The idea underlying multiple imputation is to use the observed data distribution to generate plausible values for the missing data, replacing them several times over several runs, then combining the results. The multiple imputation algorithm has three steps (Rubin, 1996). The first involves specifying and generating plausible values for missing values in the data. This stage, called imputation, creates multiple imputed data sets (m of them). In the second step, a statistical analysis is performed on each of the m imputed data set to estimate quantities of interest. The results of the m analyses will differ because the m imputations differ. There is variability both within and between the imputed data set because of the uncertainty related to missing values. The third step pools the m estimates into one, combining both within- and between- imputation variation.

Several authors have addressed the question of the optimal number of imputations. Rubin recommended 2 to 5 imputations in (Rubin, 1987). He argued that even with 50% missing data, five imputed data sets would produce point estimates that were 91% as efficient as those based on an infinite number of imputations. In 1998, Graham *et al.* (2007) suggested 20 or more imputations. Later Bodner (2008) and White *et al.* (2011) suggested the rule of thumb that m, the number of imputations, should be at least equal to the percentage of missing entries, which is what we do in this paper.

Multiple imputation by chained equations (MICE) is a practical approach to generating imputation in the first step of multiple imputation (Van Buuren and Groothuis-Oudshoorn, 2011). A more detailed description of the theory involved is provided by Van Buuren (2007), Van Buuren and Groothuis-Oudshoorn (2011), and Azur *et al.* (2011). In this study, we used the `mice` package (Van Buuren, 2018).

9.2.4.2 K-Nearest Neighbours imputation

The method of K-Nearest Neighbours imputation estimates a missing data point using values calculated from its K nearest neighbours, defined in terms of similarity (Dixon, 1979). In particular, Nearest Neighbours imputation can be with respect to some distance function.

Types of distances that can be used include the Pearson correlation, Euclidean, Mahalanobis, Chebyshev, and Gower distances. Typically, two far apart vectors are less likely than close together ones to have similar values. For a given missing data point, KNNimpute searches the whole data sets for its nearest neighbours. The missing value is then replaced by averaging the (non-missing) values of these neighbours. The method's accuracy depends on the number of neighbours taken into account. The Gower distance is coded by Kowarik and Templ (2016) in the `VIM` R package (Templ *et al.*, 2017).

9.2.4.3 Singular Value Decomposition imputation

Troyanskaya *et al.* (2001) proposed The Singular Value Decomposition imputation algorithm. This algorithm estimates missing values as linear combinations of the k most significant eigen-

vectors, where the most significant eigenvector is the one with the largest, in absolute value, eigenvalue. In this study, we used the `bcv` package (Perry, 2015) to run this.

9.3 Simulation procedure

9.3.1 Reference data set construction

Complete data set with a defined number of components were generated using the method described in Li *et al.* (2002). The actual number of components was chosen to be 2, 4 and 6. The univariate response **y** was distributed according to a Gaussian distribution $\mathcal{N}(0,1)$. Simulations were performed by adapting the `simul_data_UniYX` function available in the `plsRglm` package (Bertrand *et al.*, 2014).

9.3.2 Data dimensions

PLS regression is particularly pertinent for data matrices **X** in which $n < p$, but the behaviour of the NIPALS algorithm can depend on whether **X** in which $n < p$, or vice versa. Its properties have thus been studied on vertical data matrices, i.e., those for which $n > p$ (e.g., n = 100 and p = 20 in our study) and horizontal data matrices, i.e., those for which $n < p$ (e.g., n = 20 and p = 100 here). The range of scenarios we consider is shown in Section 9.3.4.

9.3.3 Missing data and missingness mechanism

Missing data were generated under MCAR and MAR. Missing data are simulated only on matrix **X**. The percentage of missing data d took values in $d \in \{5, 10, \ldots, 50\}\%$. Smaller proportions than 5% of missing data could have been used for $n = 100$, but preliminary runs showed that the results were very close to those for $d = 5\%$. Moreover, it was decided not to include missing data rates larger than 50% in our study since it is more than questionable to run a model on a data set in which more than half of the data are missing.

9.3.4 Simulation study design

The simulation study (see Figure 9.1) was designed as follows :

1. Data were simulated as in Li *et al.* (2002) on the univariate response **y** and on outcome variable **X** with n (number of observations) and p (number of variables) and t^* (the actual number of components) set to 2, 4 and 6 with each of the following six dimensions set-ups :
   - $n = 500$ and $p = 20$,
   - $n = 100$ and $p = 20$,

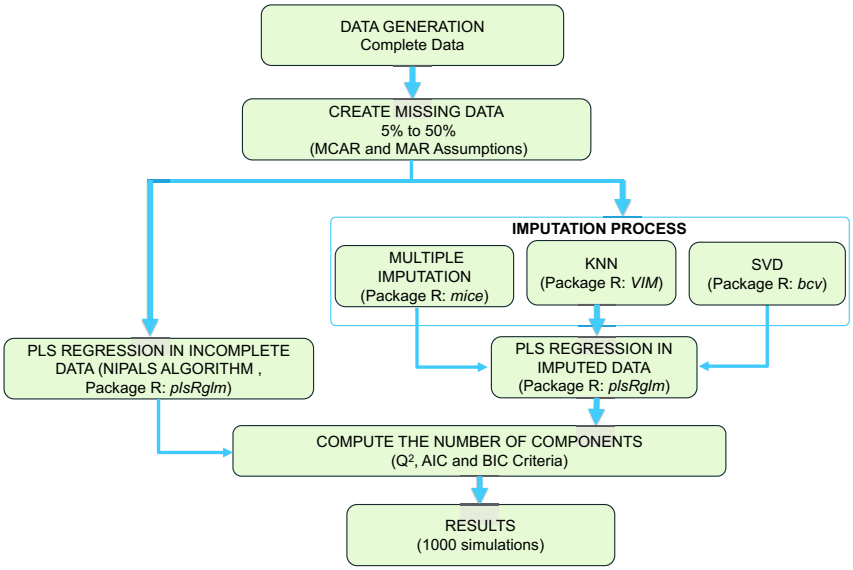


Figure 9.1 – Simulation design.

- $n = 80$ and $p = 25$,
- $n = 60$ and $p = 33$,
- $n = 40$ and $p = 50$,
- $n = 20$ and $p = 100$.

2. Missing data were created in the $\mathbf{X}$ matrices under the MCAR and MAR assumptions, with the proportion of missing data going from 5% to 50% in steps of 5%.

3. Missing data were imputed using MICE from the `mice` package and the `norm` imputation method, KNNimpute with the `VIM` package, and SVDimpute with the `bcv` package.

4. The number of components was computed using Q^2 leave-one-out cross-validation and Q^2 10-Fold cross-validation computed on the incomplete data according to the *standard* or *adaptative* methods in the `PlsRglm` package. In the multiple imputation, the number of components was calculated—by Q^2 cross-validation—as the mode of the computed number of components across all m imputations, where m was equal to the percentage of missing data (White *et al.*, 2011).

For each combination of (i) proportion of missing data, (ii) matrix dimensions, and (iii) type of missingness, 1000 replicate data set were drawn.

9.4 Simulation results

9.4.1 Complete data set

First, the complete data set were simulated related to each data set (Table 9.1). We can see that the Q^2-10-Fold criterion is the best criterion as it selects the largest true number of components in

every dimension either MCAR or MAR assumptions. When t^*= 2, the correct model dimension was selected in 97% cases. Furthermore, the performances of both Q^2-LOO and Q^2-10-Fold criteria for selecting the correct number of components decrease when the sample size decreases, the number of variables increases and the number of components increases. Similar conclusions can be drawn for AIC-DoF and BIC-DoF. These methods, however, perform less well and, overall, selects the correct number of components less frequently than Q^2 does.

On the other hand, the AIC and BIC criteria are the less efficient ones to determine the actual number of components. It generally yields a larger number of components than expected. The AIC and BIC criteria selected eight components in almost every run of the 1000 simulations (results not shown here). It must be reminded that the number of computed components was set to a maximum of eight. Thus it cannot be excluded that the observed number of components may have been larger in some of the simulations.

Table 9.1 – The evaluation of complete data. The results are expressed as number of simulations for which the selected components number (t^*) equals to 2, 4 and 6 (the actual value) over 1000 simulations, n is the number of observations, p is the number of variables.

n	p	$t^*=2$						$t^*=4$						$t^*=6$					
		Q^2-LOO	Q^2-10-Fold	AIC	AIC-DoF	BIC	BIC-DoF	Q^2-LOO	Q^2-10-Fold	AIC	AIC-DoF	BIC	BIC-DoF	Q^2-LOO	Q^2-10-Fold	AIC	AIC-DoF	BIC	BIC-DoF
20	100	469	970	0	0	0	0	190	724	0	0	0	0	49	134	0	0	0	0
40	50	896	986	0	161	0	259	817	925	0	297	0	377	770	786	0	392	0	515
60	33	979	995	0	483	0	786	945	958	0	508	0	688	958	944	0	466	0	644
80	25	991	998	0	689	0	886	977	977	0	630	0	792	987	987	0	484	0	662
100	20	996	999	0	767	0	905	992	994	0	708	1	846	995	997	0	389	4	572
500	20	1000	1000	0	884	7	951	1000	1000	0	840	12	920	1000	1000	0	128	29	391

9.4.2 Comparison of the different algorithms

The framework for our simulations was based on those in Li *et al.* (2002) with the actual number of components is set to 2, 4 and 6. In detail, Figures 9.3 to 9.14 plot the performance of each method as a function of the proportion of missing data with various shaped matrices under both MCAR and MAR. A summary of these results for all criteria and methods are shown in Tables 9.2.

These results show that the performance of Q^2-LOO in selecting the correct number of components increases generally as the sample size does, and as expected, decreases as the proportion of missing data increases for both MCAR and MAR. The NIPALS-PLSR with Q^2-LOO generally provides a satisfactory performance with any dimension under MCAR for $t^* = 2$. In comparison, the MICE with Q^2-LOO, Q^2-10-Fold perform well under both MCAR and MAR assumptions for $t^* = 4$ and $t^* = 6$. They give results closest to the correct number of components on the incomplete data set when the proportion of missing data was small ($< 30\%$).

The SVDimpute with Q^2-LOO performs the worst when $t^* = 4$ and $t^* = 6$. The actual number of components was correctly selected in only around one-third of the simulations. On the contrary, it works well when $t^* = 2$ for horizontal data matrices setting under MAR assumption and the proportion of missing data equals 5%.

We see that the Q^2-10-Fold performs less well and, overall, selects the correct number of components less frequently than Q^2-LOO does. In the vast majority of situations (i.e., combinations of matrix size, proportion, and pattern of missing data), the number of components selected is on average larger than the actual number of components.

We also found that AIC, AIC-DoF, and BIC systematically select a larger number of components than Q^2. This difference can sometimes be as large as three or four for each of the methods and both the MCAR and MAR cases.

The true number of components selected by either MICE, KNNimpute, or SVDimpute with the BIC-DoF criterion are systematically larger than the number of components selected by BIC, AIC, and AIC-DoF. BIC-DoF's performance increases and then decreases as the proportion of missing data increases, instead of regularly decreasing over the whole range of missing data proportions.

9.5 Real data

We applied PLS regression to four type data : (i) The bromhexine data in a pharmaceutical syrup (Goicoechea and Olivieri, 1999a) ; (ii) The tetracycline data in serum (Goicoechea and Olivieri, 1999b) ; (iii) The Los Angeles ozone pollution data 1976 which is provided by the `mlbench` package (Leisch and Dimitriadou, 2010) ; (iv) The octane data in gasolines from NIR data (Goicoechea and Olivieri, 2003).

The focus of this paper is to determine the number of components of a PLS regression fitted with the NIPALS algorithm. The real data analysis aims to extract components to build representative models of the process. We repeatedly selected one hundred times the number of components in complete data using the $q = 10$ CV-criterion. The larger number of components that could be selected was twelve. The results of this selection process are displayed in Figure 9.2 with a summary in Table 9.3. The selected significant number of components in complete data real (t^{**}) in Bromhexine, Tetracycline, Los Angeles ozone pollution, and octane data set is 3-, 5-, 2- and 2-components, respectively. Besides, in Bromhexine data, $t^{**}= 2$ can be selected based on this number of components being selected in 42 out of 100 runs.

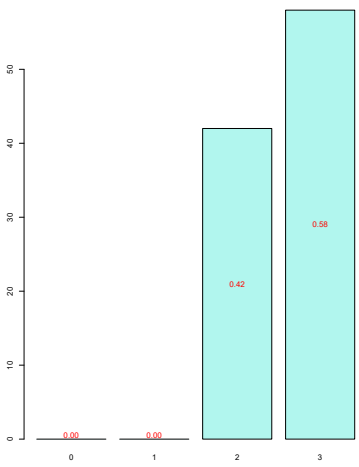
The procedure for real data analysis is the same as a simulation study. Missing data were first created on complete real data, both MAR and MCAR assumptions from 5% to 50% in steps 5%. PLS regression was applied then on incomplete data with NIPALS-PLSR and PLS regression on imputed data set which used three methods of imputation : multiple imputation by chained equations (MICE), k-nearest neighbour imputation (KNNimpute) and a singular value decomposition imputation (SVDimpute). Finally, the number of components is computed using Q^2, AIC and BIC and their performance are compared with t^{**}. For addition, we also compare the performance of real data results with the simulation results. The real data evaluation of the criteria for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR and MAR assumptions with different proportions of missing values are shown in Tables 9.5 – 9.8. A summary of the real data results for all criteria is shown in Table 9.4.

9.5.1 Bromhexine data

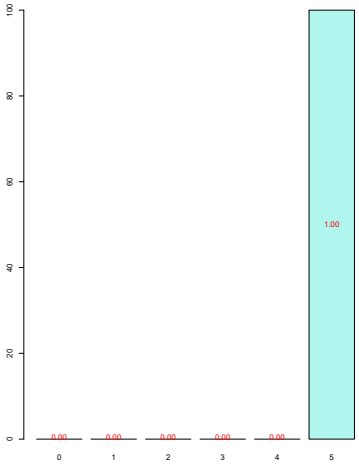
In this data set, the author discussed the possibility of determining bromhexine in syrups by applying electronic absorption measurements together with robust multivariate calibration

Table 9.2 – The evaluation of NIPALS-PLSR, MICE, KNNimpute and SVDimpute. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 2, 4 and 6 (the true value), n is the number of observations, p is the number of variables.

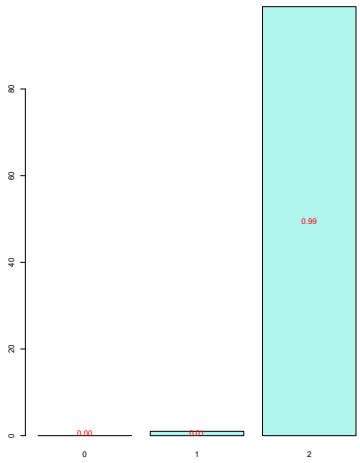
n	p	Assumption	$t^* = 2$						$t^* = 4$						$t^* = 6$					
			Q^2-LOO	Q^2 -10-Fold	AIC	AIC-DoF	BIC	BIC-DoF	Q^2-LOO	Q^2 -10-Fold	AIC	AIC-DoF	BIC	BIC-DoF	Q^2-LOO	Q^2 -10-Fold	AIC	AIC-DoF	BIC	BIC-DoF
20	100	MCAR	NIPALS-PLSR	MICE KNNimpute SVDimpute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		MAR		KNNimpute SVDimpute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	50	MCAR	NIPALS-PLSR MICE	MICE KNNimpute SVDimpute	-	-	-	-	MICE	MICE	-	-	-	-	-	- -	-	-	-	-
		MAR	KNNimpute SVDimpute	KNNimpute SVDimpute MICE	-	-	-	-	MICE	MICE	-	-	-	-	-	-	-	-	-	-
60	33	MCAR	NIPALS-PLSR KNNimpute	KNNimpute SVDimpute MICE	-	-	-	-	MICE KNNimpute NIPALS-PLSR	MICE KNNimpute	-	-	-	-	-	-	-	-	-	-
		MAR	KNNimpute SVDimpute	KNNimpute SVDimpute	-	-	-	-	MICE KNNimpute	MICE	-	-	-	-	-	-	-	-	-	-
80	25	MCAR	NIPALS-PLSR KNNimpute	KNNimpute SVDimpute	-	-	-	-	MICE NIPALS-PLSR KNNimpute	MICE KNNimpute	-	MICE	-	MICE	MICE	MICE	-	-	-	-
		MAR	KNNimpute NIPALS-PLSR	KNNimpute SVDimpute	-	-	-	-	MICE NIPALS-PLSR KNNimpute	MICE	-	MICE	-	MICE	MICE	MICE	-	-	-	-
100	20	MCAR	NIPALS-PLSR KNNimpute MICE	KNNimpute MICE SVDimpute	-	-	-	MICE	MICE NIPALS-PLSR KNNimpute	MICE KNNimpute	-	MICE	-	MICE	MICE	MICE	-	-	-	-
		MAR	KNNimpute MICE NIPALS-PLSR	KNNimpute MICE SVDimpute	-	-	-	MICE	MICE KNNimpute NIPALS-PLSR	MICE KNNimpute	-	MICE	-	MICE	MICE	MICE	-	-	-	-
500	20	MCAR	NIPALS-PLSR KNNimpute MICE	KNNimpute MICE	-	-	-	MICE	NIPALS-PLSR MICE KNNimpute	MICE KNNimpute	-	MICE	-	MICE	MICE KNNimpute NIPALS-PLSR	MICE KNNimpute	-	-	-	-
		MAR	NIPALS-PLSR KNNimpute MICE	KNNimpute MICE	-	-	-	MICE	NIPALS-PLSR MICE KNNimpute	MICE KNNimpute	-	MICE	-	MICE	MICE KNNimpute NIPALS-PLSR	MICE KNNimpute	-	-	-	-



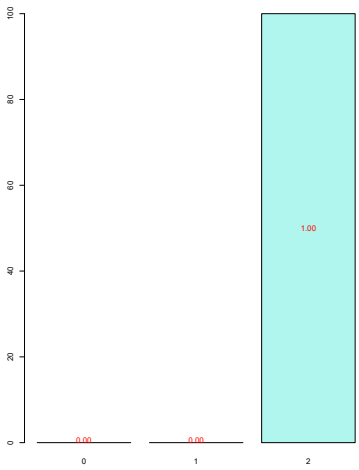
(a) Bromhexine data set



(b) Tetracycline data set



(c) Los Angeles ozone pollution data set



(d) Octane data set

Figure 9.2 – Plot of extracted significant numbers of components in complete data real using Q^2-10-Fold with 100 times.

Table 9.3 – Selected significant number of components in complete data real (t^{**}).

Data set	Bromhexine	Tetracycline	Los Angeles ozone pollution	Octane
t^{**}	3(58/100) 2(42/100)	5(100/100)	2 (100/100)	2(100/100)

analyses. The authors used PLS1, among other methods. A data set of 23 samples were prepared with 64 concentrations of bromhexine (n= 23 $\times$ p= 64).

The performance of criteria for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR and MAR assumptions with different proportions of missing values are shown in Tables 9.5. For comparison, we select the dimension $n = 20 \times p = 100$ with t^*= 2 in the simulation results.

Table 9.4 – The evaluation of NIPALS-PLSR, MICE, KNNimpute and SVDimpute in incomplete data real. The results are expressed as the selected combination between the criteria and the methods that the number of components of a PLS regression is close to the selected significant number of components (t^{**}).

Data set	Dimension (n × p)	t^{**}	Assumption	Criteria					
				Q^2-LOO	Q^2 -10-Fold	AIC	AIC-DoF	BIC	BIC-DoF
Bromhexine	23 x 64	3 / 2	MCAR	NIPALS-PLSR SVDimpute	KNNimpute SVDimpute	-	-	-	-
			MAR	NIPALS-PLSR SVDimpute	SVDimpute	-	-	-	-
Tetracycline	107 x 101	5	MCAR	SVDimpute	-	-	-	-	-
			MAR	SVDimpute	-	-	-	-	-
Los Angeles ozone pollution	203 x 12	2	MCAR	MICE KNNimpute	MICE KNNimpute	-	-	-	-
			MAR	MICE KNNimpute	MICE KNNimpute	-	-	-	-
Octane	68 x 493	2	MCAR	KNNimpute NIPALS-PLSR SVDimpute	KNNimpute MICE SVDimpute	-	-	-	-
			MAR	KNNimpute NIPALS-PLSR SVDimpute	KNNimpute - -	-	-	-	-

The SVD impute, with either Q^2-LOO or Q^2-10-Fold criterion generally performs well to select the correct number of components (t^{**}= 2 and 3) in this data set, under both MCAR and MAR assumptions. These situations also correspond with the simulation results.

In contrast, the performance of the AIC, AIC-DoF, BIC, and BIC-DoF include too many components, clearly overfitting the data in this case. In most of the results, the selected number of components is almost five times the selected significant number of components (10 components). This finding is also supported by the results that we obtained with our previous simulation study. We found that the numbers of components selected by those criteria were systematically larger than the correct number of components.

9.5.2 Tetracycline data set

Tetracycline has been determined in human serum samples. It has 107 observations on 101 explanatory variables (n= 107 × p= 101) with spectral range (in nm) between 0 and 600. The tetracycline data set was obtained from Richet Laboratories, and its purity was checked according to Pharmacopeia recommendations. This research discussed the possibility of quantitating tetracycline and its derivatives in blood-serum samples by applying synchronous spectrofluorometric measurements together.

The overall performance of the criteria for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR and MAR assumptions with different missing data proportions are shown in Tables 9.6. We did not compare this data set with the simulation results because we could not correctly match the data sets dimensions.

Either for the MCAR or MAR situation, the performance of SVDimpute with Q^2-LOO criterion is the best combination between the methods and the criteria to determine the selected significant number of components (Table 9.3). As with the case of Bromhexine data set, the AIC, AIC-DoF, BIC, and BIC-DoF criteria have less good performances overall. They select a larger number of component than the Q^2-LOO and Q^2-10-Fold criteria. The selected dimension can sometimes be twice the correct dimension.

9.5.3 Los Angeles ozone pollution data set

Los Angeles ozone pollution data set concerns 12 predictor variables which contain the measurement dates and, among others, information on wind speed, humidity, temperature. This data set contains 366 daily observations used to predict the daily maximum one-hour-average ozone reading. The original data contains missing values. Among the 366 samples, we used the 203 complete ones (n= 203 $\times$ p= 12).

The evaluation of the criteria for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR and MAR assumptions with different proportions of missing values are shown in Tables 9.7. We compared the real data set results (t^{**}= 2) with the simulation results on $n = 100 \times p = 20$ and t^{*}= 2 that almost matched the real data set's dimensions.

In the Los Angeles ozone pollution data set analysis, the number of selected components using Q^2-LOO and Q^2-10-Fold criteria on MICE and KNNimpute are the closest, for any proportion of missing data, to the number of components that we selected using complete data. They perform well to determine t^{**}= 2. These results correspond with the simulation results obtained under the MAR assumption but are somewhat different from those obtained under the MCAR assumption. The NIPALS-PLSR and KNNImpute perform well in the simulation.

Generally, AIC, AIC-DoF, BIC, and BIC-DoF have less good performance. These criteria include too many components. The selected number of components are almost more than twice as large as the selected significant number of components. This finding is also supported by the simulation results obtained.

9.5.4 Octane data set

The experiment of Octane data set studied the concentration of glucuronic acid in complex mixtures studied by Fourier transform mid-infrared spectroscopy and the octane number in types of gasoline monitored by near-infrared spectroscopy. Determination of octane in types of gasoline from NIR data on this study used 68 NIR spectra of gasoline samples collected in a local distillery, in the range 4020 - 9996 cm^{-1} which is categorized in 493 explanatory variables (n= 68 $\times$ p= 493).

The evaluation of the criteria for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute both MCAR and MAR assumptions with different proportions of missing values are shown in (Table 9.8).

All methods except MICE with Q^2-LOO criterion exhibit good and similar performance in

terms of determination of the number of components, whatever the proportion and mechanism of missing data. The AIC, AIC-DoF, BIC, and BIC-DoF include too many components. The difference can sometimes be as large as eight components.

9.6 Discussion and conclusion

PLS regression is a multivariate method for which two algorithms (SIMPLS and NIPALS) can be used to provide the model's parameter estimates. The NIPALS algorithm has the interesting property of being able to provide estimates on incomplete data ; this has been extensively studied in the case of PCA—for which NIPALS was originally devised.

Here, we have studied the behavior of the NIPALS algorithm when used to fit PLS regression models for various proportions of missing data and different types of missingness. Comparisons with MICE, KNNimpute, and SVDimpute were performed under the MCAR and MAR assumptions. In our simulations, the model dimension (i.e., the optimal number of components) was computed according to different criteria, including Q^2, AIC, and BIC.

The number of selected components, be it for complete, incomplete, or imputed data set, depends on the criterion used. The fact that AIC and BIC select a larger number of components than Q^2 has been already observed in another context by the present authors (Meyer *et al.*, 2010) and by others (Li *et al.*, 2002).

In the simulation results of Q^2-LOO, the number of components selected using NIPALS-PLSR (two-components) under MCAR assumption and MICE (two- and four-components) under MCAR and MAR assumptions are much closer to the correct number of components when the proportion of missing data is small ($< 30\%$) and for vertical matrices. However, the MICE computation time was long, and depended on the proportion of missing data, increasing as the proportion of missing data did. For instance, the average MICE run time for $n = 100 \times p = 20$, was about 11 times longer than that of NIPALS-PLSR when $d = 10\%$, and around 40 times longer when $d = 50\%$ under MCAR. In contrast, NIPALS-PLSR, KNNimpute, and SVDimpute had swift run times : 0.5–1.5 seconds on average. Generally, the run time under MAR was longer than under MCAR for both vertical and horizontal matrices. Consequently, though MICE may be the method of choice, its run time may prohibit its use in practice.

BIC-DoF, a criterion derived from BIC, gives a slightly better estimation of the number of components in the simulation when the proportion of missing data is small, particularly for MICE. This finding shows that taking into account a modified number of DoF can substantially improve the likelihood of selecting the correct number of components (Krämer and Sugiyama, 2011). Further research would nevertheless be useful to extend this version of BIC to other settings like, for instance, GLM or adapt it to specific cases of incomplete data set that require further *DoF* adjustments. In contrast to the real data results, the performance of BIC-DoF includes too many components.

For smaller sample sizes n, the multivariate structure of the data was not taken into account in the imputations due to high levels of collinearity. Indeed, the smaller the sample size, the more difficult it was for the MICE algorithm to converge. Thus, though it would have been possible to run the imputation, the PLS regression estimates would have been biased. This result implies

that our conclusions for tiny sample sizes may be misleading. Such biased parameter estimates could also bias the comparisons between the methods but also hint at the fact that even a small proportion of missing data can make it difficult to estimate the correct number of components in PLS regression.

In the vast majority of situations, either the simulation results or the practical examples discussed (that is any combination of size, proportion and pattern of missing data), It is clear that the Q^2-LOO criterion has the best performance. Theoretically, the leave-one-out method can extract the maximum possible information (Eastment and Krzanowski, 1982). The simulation results in this study support this result. Furthermore, our real data analysis shows similar results and our simulation results back up our real data analysis.

In conclusion, our simulations show that whatever the criterion used, the type of missingness and proportion of missing data must also be taken into consideration since they both influence the number of components selected. These results match our real data studies. The actual number of components of a PLS regression was challenging to determine, especially for small sample sizes, and when the proportion of missing data was larger than 30%. Moreover, under MCAR, the number of selected components using these methods was generally closer to the actual number of components than in the MAR setting.

Table 9.5 – Bromhexine data set : the results are the number of components selected for each criteria, *d* is the percentage of missing value.

Assumption	d	Q^2-LOO					Q^2-10-Fold				AIC				AIC-DoF			BIC				BIC-DoF		
		NIPALS-PLSR (standard)	NIPALS-PLSR (adaptative)	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	MICE	KNNimpute	SVDimpute
MCAR	5	3	3	4	4	2	NA	1	2	2	10	7	10	8	9	10	10	10	7	10	5	9	10	10
	10	2	2	1	4	2	NA	1	2	2	10	6	10	5	1	10	9	10	5	10	5	1	10	9
	15	2	2	1	1	2	NA	1	1	1	10	2	10	3	1	10	4	10	2	10	3	1	10	4
MAR	5	4	4	4	2	2	NA	1	2	2	10	10	10	5	10	10	6	10	10	10	5	10	10	6
	10	3	3	1	1	2	NA	1	1	2	10	7	10	3	9	10	8	10	7	10	3	9	10	8
	15	3	3	1	1	1	NA	1	1	1	10	3	10	2	4	10	1	10	3	10	2	1	10	1

Table 9.6 – Tetracycline data set : the results are the number of components selected for each criteria, *d* is the percentage of missing value.

Assumption	d	Q^2-LOO					Q^2-10-Fold				AIC				AIC-DoF			BIC				BIC-DoF		
		NIPALS-PLSR (standard)	NIPALS-PLSR (adaptative)	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	MICE	KNNimpute	SVDimpute
MCAR	5	5	5	5	5	5	3	5	4	5	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	10	5	5	5	5	5	3	5	4	5	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	15	3	3	5	5	5	3	5	4	5	10	10	10	10	5	10	10	10	10	10	10	5	10	10
	20	5	5	5	5	5	3	5	4	5	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	25	5	5	5	4	5	3	5	3	4	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	30	5	5	3	4	5	3	3	4	4	10	10	10	10	3	7	10	10	10	10	10	3	6	10
	35	3	3	3	4	5	3	3	3	4	10	10	10	10	3	10	10	10	10	10	10	3	10	10
	40	5	5	3	4	5	3	2	3	5	10	10	10	10	8	10	10	10	10	10	10	8	4	10
	45	5	5	3	3	5	3	2	3	4	10	10	10	7	3	10	10	10	10	10	7	2	10	10
	50	4	4	2	3	5	3	2	3	4	10	10	10	10	2	10	10	10	10	10	10	2	10	10
MAR	5	5	5	5	5	5	3	5	5	5	10	10	10	10	10	10	10	10	10	10	10	10	10	5
	10	5	5	5	5	5	3	5	4	5	10	10	10	10	10	10	9	10	10	10	10	10	10	8
	15	5	5	5	5	5	3	5	4	5	10	10	10	10	5	9	10	10	10	10	10	5	9	10
	20	5	5	5	5	5	3	5	4	5	10	10	10	10	10	6	10	10	10	10	10	10	5	10
	25	5	5	5	4	5	3	4	4	5	10	10	10	10	10	6	10	10	10	10	10	5	4	10
	30	5	5	3	4	5	3	3	3	4	10	10	10	10	10	10	10	10	10	10	10	3	4	10
	35	3	3	3	3	5	3	2	3	4	10	10	10	10	5	10	10	10	10	10	10	2	10	10
	40	5	5	3	3	5	3	2	3	4	10	10	10	8	10	10	10	10	10	10	8	2	3	10
	45	4	4	1	3	4	4	1	3	4	10	10	10	10	2	10	10	10	10	10	10	1	10	10
	50	4	4	1	3	5	4	1	3	4	10	10	10	10	8	4	10	10	10	10	6	1	3	10

Table 9.7 – Los Angeles ozone pollution data set : the results are the number of components selected for each criteria, *d* is the percentage of missing value.

Assumption	d	Q^2-LOO					Q^2-10-Fold				AIC				AIC-DoF			BIC				BIC-DoF		
		NIPALS-PLSR (standard)	NIPALS-PLSR (adaptative)	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	MICE	KNNimpute	SVDimpute
MCAR	5	2	2	2	2	2	1	2	2	2	5	5	5	4	9	10	7	4	4	4	4	7	4	7
	10	2	2	2	2	2	1	2	2	2	7	7	5	5	10	9	6	4	4	5	4	7	6	7
	15	2	2	2	2	2	1	2	2	1	7	5	7	4	7	7	4	4	4	3	4	7	2	6
	20	2	2	2	2	2	1	2	2	2	7	5	5	4	9	7	5	7	4	4	4	7	5	7
	25	2	2	2	2	1	1	2	2	1	6	5	6	4	9	9	5	4	4	3	4	7	5	5
	30	2	2	2	2	2	1	2	2	1	7	4	5	4	7	8	10	4	4	4	4	8	5	4
	35	2	2	2	2	2	1	2	2	1	7	5	5	3	7	7	4	5	4	3	4	4	4	7
	40	1	1	2	2	1	1	2	2	1	6	4	5	3	4	7	4	4	4	3	3	7	2	4
	45	2	2	2	2	1	1	2	2	1	5	6	4	3	9	9	7	4	4	2	5	4	2	9
	50	2	2	2	2	1	1	2	2	1	5	5	4	3	7	6	5	4	4	3	4	4	2	4
MAR	5	2	2	2	2	2	1	2	2	2	5	5	5	4	9	9	6	4	4	4	4	7	7	4
	10	2	2	2	2	2	1	2	2	1	7	5	5	4	10	9	4	4	4	4	4	8	7	4
	15	2	2	2	2	2	1	2	2	2	8	5	5	4	7	7	4	4	4	4	4	7	7	4
	20	2	2	2	2	2	1	2	2	2	7	5	5	4	10	7	4	4	4	4	3	4	4	4
	25	2	2	2	2	2	1	2	2	2	6	5	5	5	10	7	10	4	4	4	4	6	6	9
	30	1	1	2	2	2	1	2	2	2	7	4	4	3	9	6	4	4	4	4	3	4	4	4
	35	2	2	2	2	2	1	2	2	1	7	6	5	3	10	7	4	5	4	4	3	7	4	2
	40	2	2	2	2	1	1	2	2	1	5	4	4	3	6	9	3	4	4	4	3	4	4	2
	45	2	2	2	2	1	1	2	2	1	5	5	5	4	7	8	7	5	4	4	3	5	8	2
	50	2	2	2	2	1	1	2	2	1	4	6	6	3	9	7	4	4	4	5	3	2	7	2

Table 9.8 – Octane data set : the results are the number of components selected for each criteria, *d* is the percentage of missing value.

Assumption	d	Q^2-LOO					Q^2-10-Fold				AIC				AIC-DoF			BIC				BIC-DoF		
		NIPALS-PLSR (standard)	NIPALS-PLSR (adaptative)	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	MICE	KNNimpute	SVDimpute	NIPALS-PLSR	MICE	KNNimpute	SVDimpute	MICE	KNNimpute	SVDimpute
MCAR	5	2	2	2	2	2	2	1	2	2	10	10	10	10	9	7	8	10	10	10	10	9	6	8
	10	2	2	4	2	2	2	2	2	1	10	10	10	10	10	10	10	10	10	10	10	10	10	3
	15	2	2	3	2	2	2	2	2	2	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	20	2	2	2	2	2	2	2	2	2	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	25	2	2	2	2	2	NA	2	2	2	10	10	10	10	10	10	10	10	10	10	10	10	10	10
MAR	5	2	2	2	2	2	2	2	2	2	10	10	10	10	10	8	10	10	10	10	10	10	8	10
	10	2	2	2	2	2	2	2	2	2	10	10	10	10	10	7	8	10	10	10	10	10	7	8
	15	2	2	2	2	2	2	2	2	2	10	10	10	10	10	5	9	10	10	10	10	10	5	9
	20	2	2	4	2	2	2	2	2	2	10	10	10	10	10	9	10	10	10	10	10	10	9	10
	25	2	2	2	2	2	NA	2	2	2	10	10	10	10	10	10	10	10	10	10	10	10	10	10

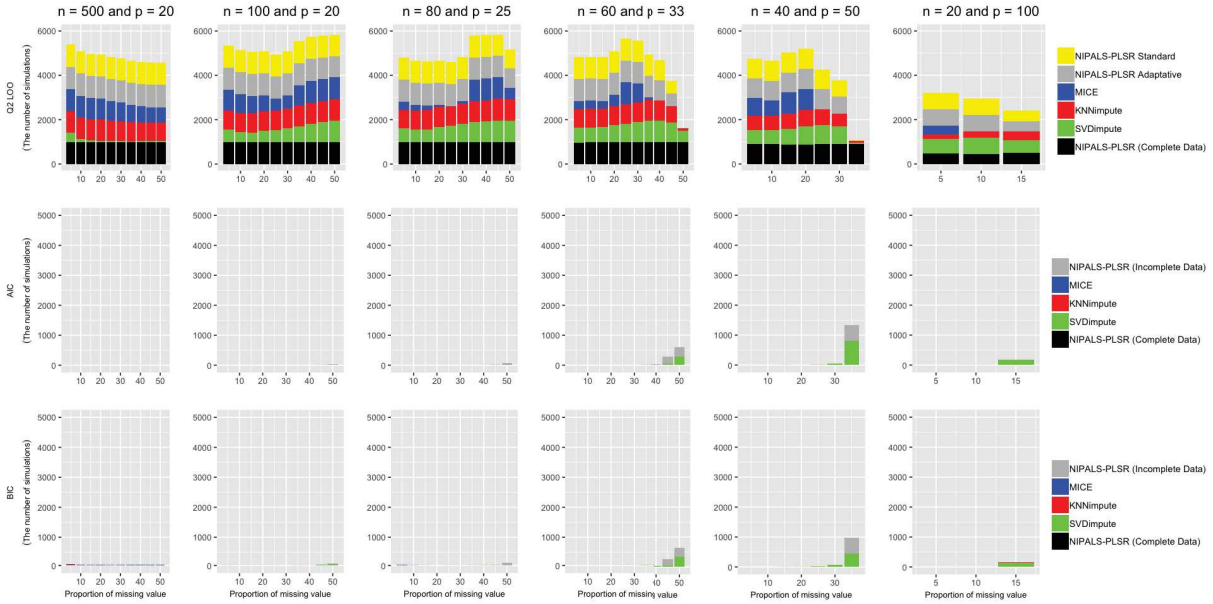


Figure 9.3 – Evaluation of Q^2-LOO, AIC, and BIC for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 2 (the true value).

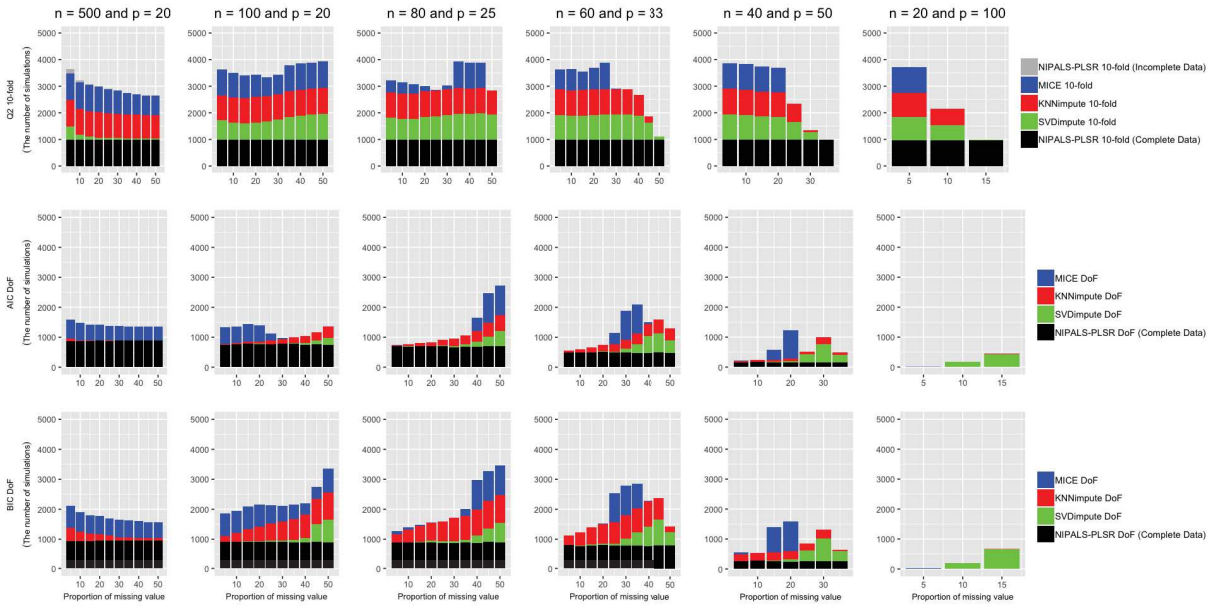


Figure 9.4 – Evaluation of Q^2-10-Fold, AIC-DoF, and BIC-DoF for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 2 (the true value).

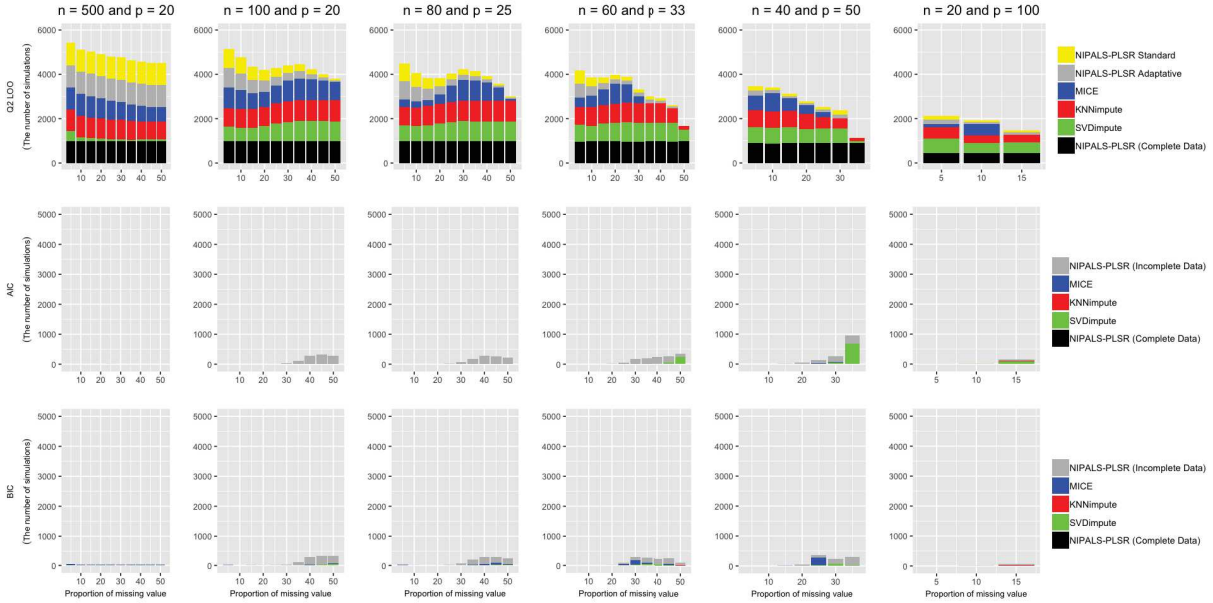


Figure 9.5 – Evaluation of Q^2-LOO, AIC, and BIC for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 2 (the true value).

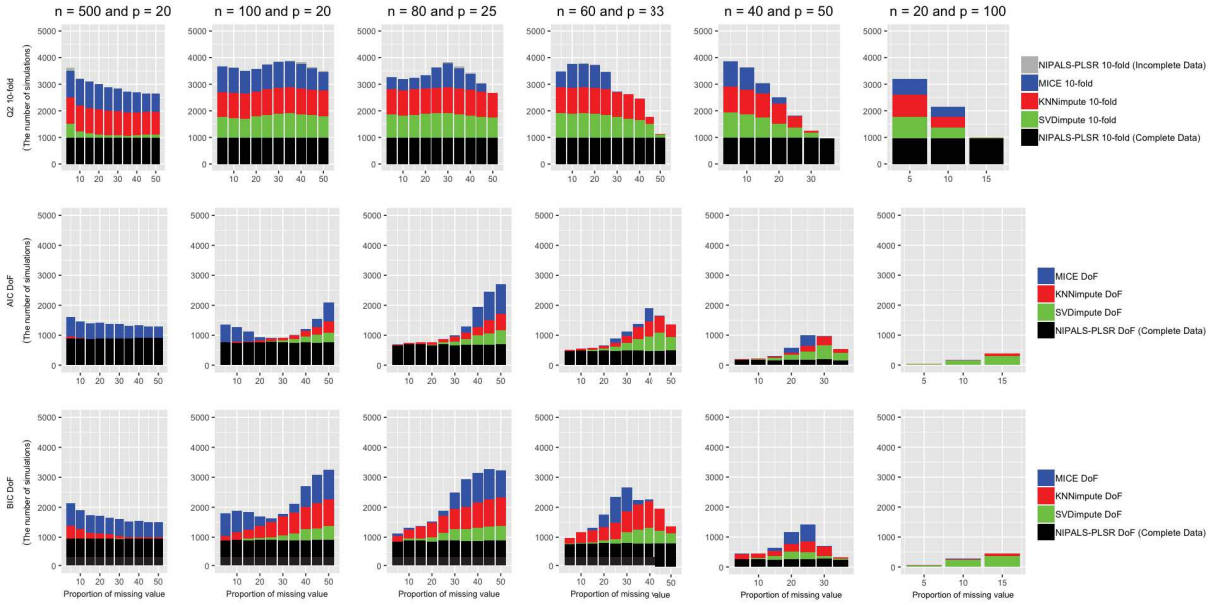


Figure 9.6 – Evaluation of Q^2-10-Fold, AIC-DoF, and BIC-DoF for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 2 (the true value).

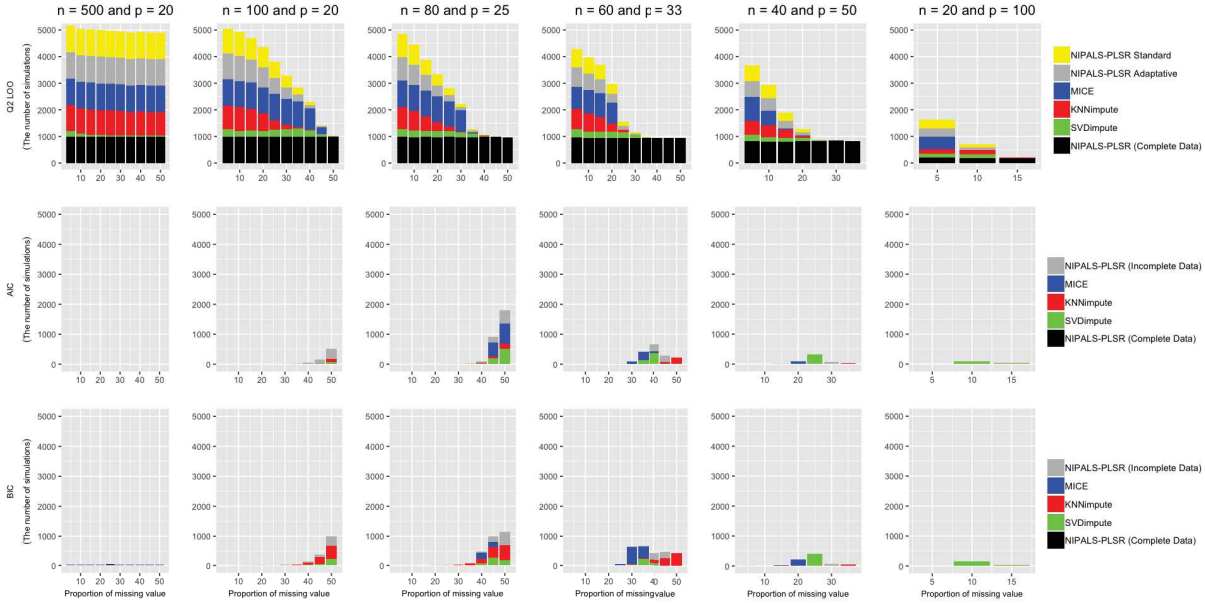


Figure 9.7 – Evaluation of Q^2-LOO, AIC, and BIC for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 4 (the true value).

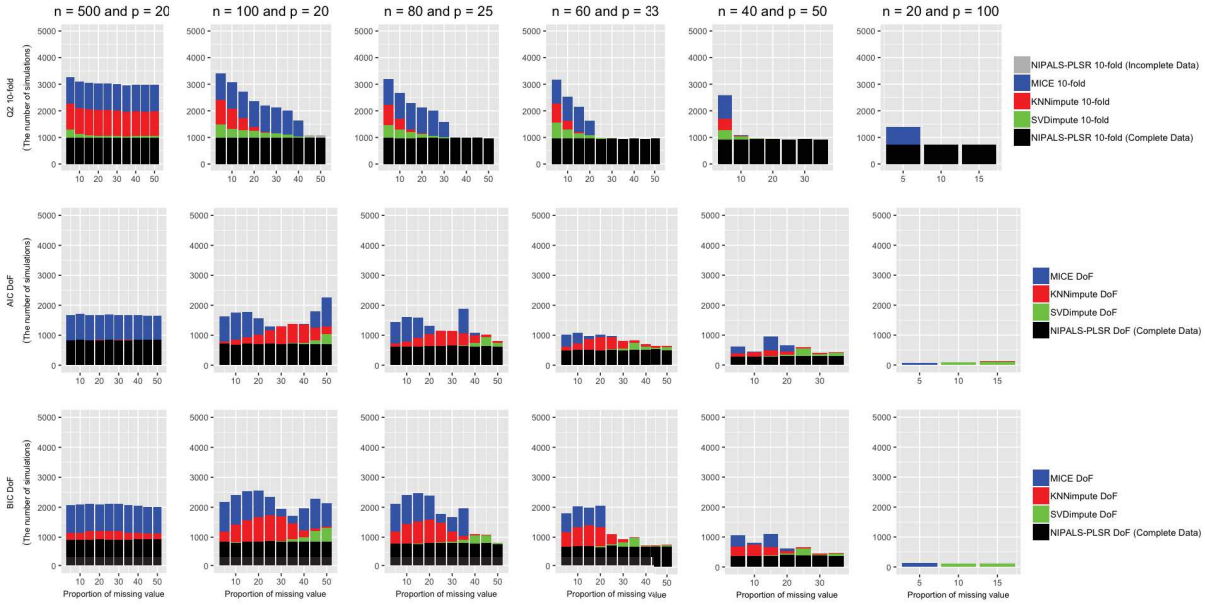


Figure 9.8 – Evaluation of Q^2-10-Fold, AIC-DoF, and BIC-DoF for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 4 (the true value).

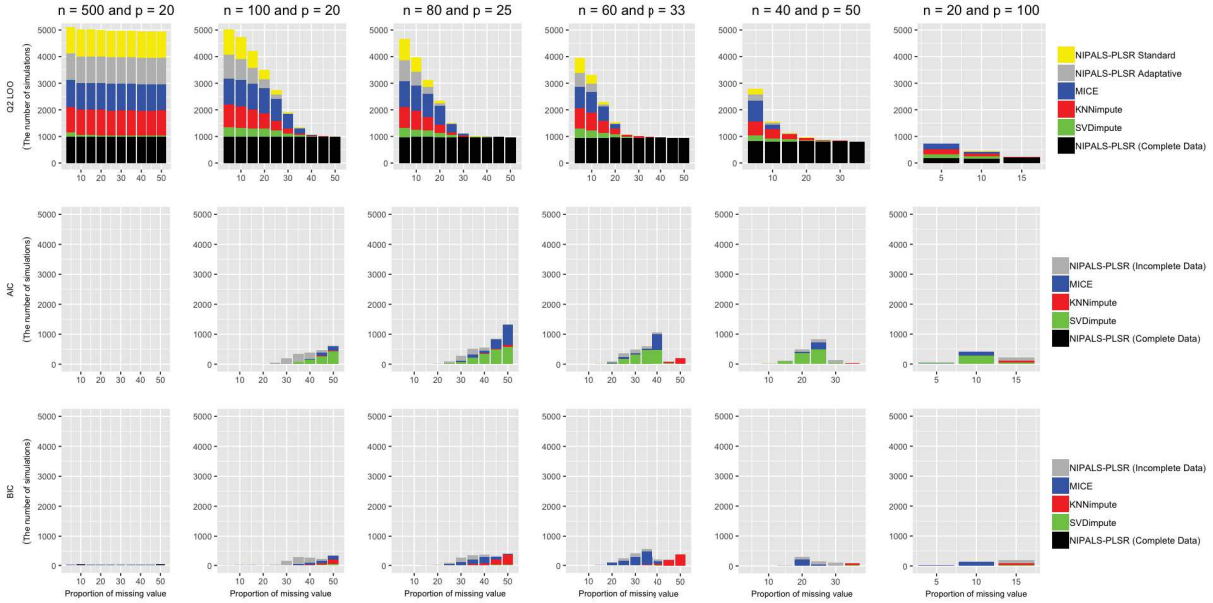


Figure 9.9 – Evaluation of Q^2-LOO, AIC, and BIC for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 4 (the true value).

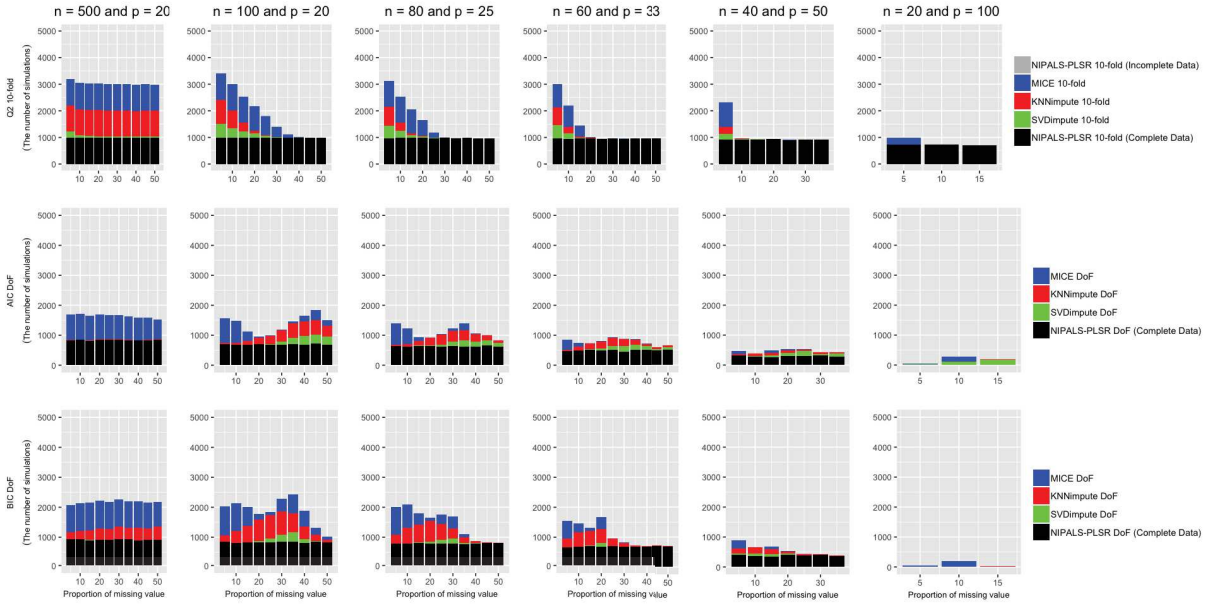


Figure 9.10 – Evaluation of Q^2-10-Fold, AIC-DoF, and BIC-DoF for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 4 (the true value).

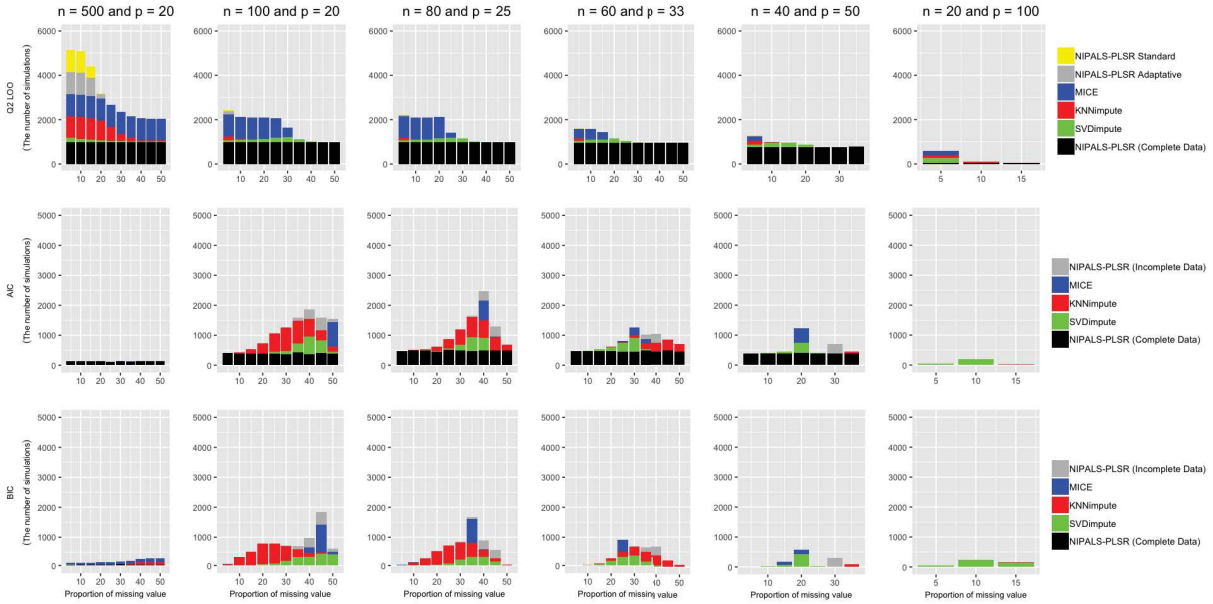


Figure 9.11 – Evaluation of Q^2-LOO, AIC, and BIC for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 6 (the true value).

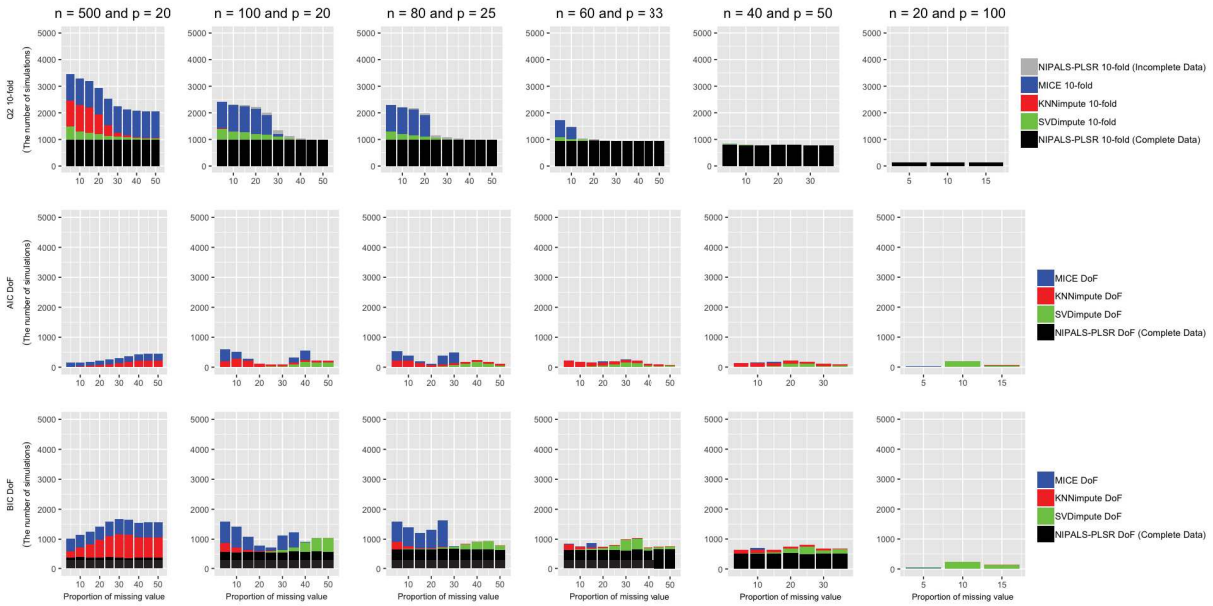


Figure 9.12 – Evaluation of Q^2-10-Fold, AIC-DoF, and BIC-DoF for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MCAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 6 (the true value).

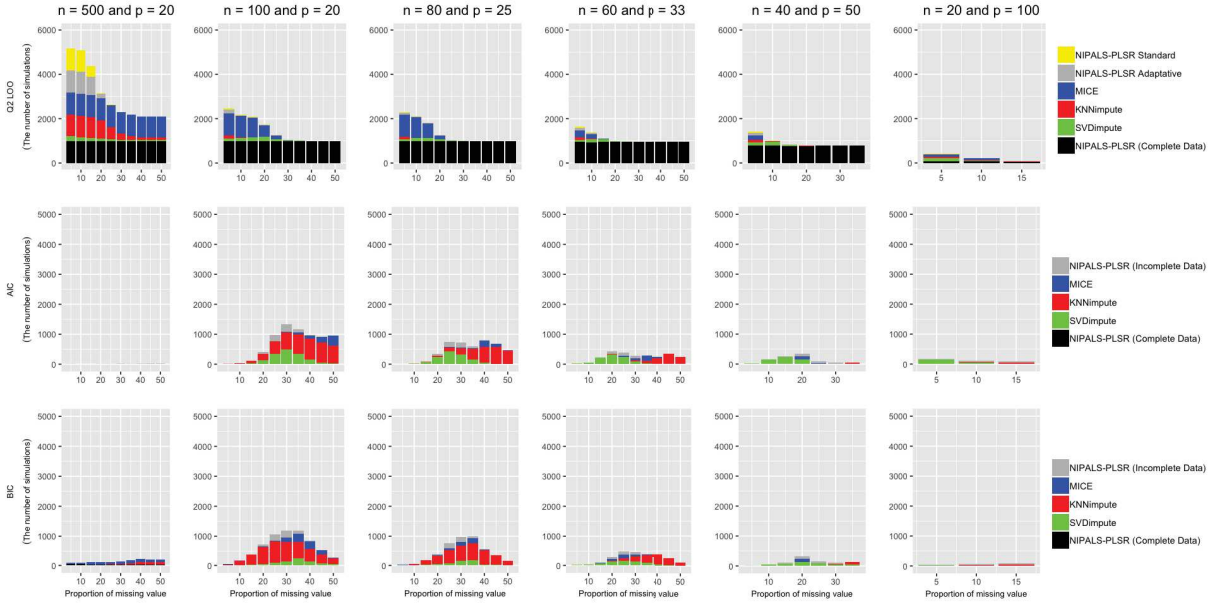


Figure 9.13 – Evaluation of Q^2-LOO, AIC, and BIC for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 6 (the true value).

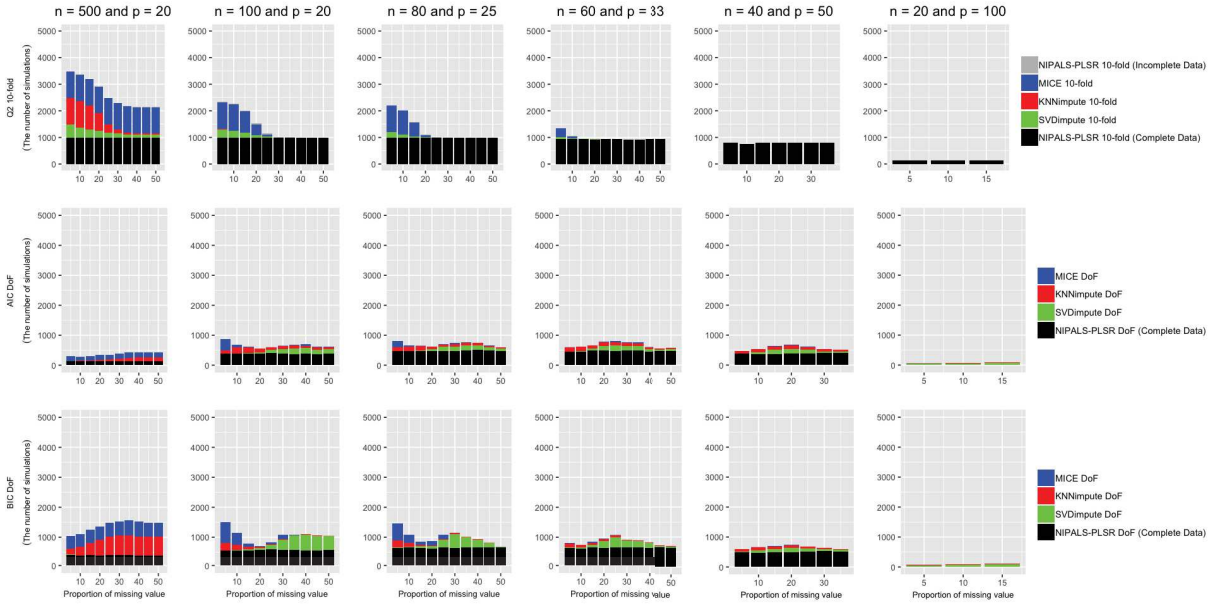


Figure 9.14 – Evaluation of Q^2-10-Fold, AIC-DoF, and BIC-DoF for NIPALS-PLSR, MICE, KNNimpute, and SVDimpute under MAR. The results are expressed as the selected combination of criteria and methods with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations for which the selected component number (t^*) equals to 6 (the true value).

Quatrième partie

Perspectives et conclusion

Chapitre 10

Vers des améliorations de la robustesse et de l'estimation de la dimension des modèles en régression *PLS*

Les éléments de ce chapitre sont en cours d'élaboration. Dans ce chapitre, nous sommes toujours intéressés par la détermination du nombre de composantes dans la régression *PLS* en examinant la relation entre une matrice de variables explicatives **X** et un variable réponse **y** sous la forme d'une combinaison d'un modèle d'l'Analyse en Composantes Principales Probabiliste (ACPP) et de régression linéaire. La régression *PLS* probabiliste permet d'établir cette relation par un modèle qui a été proposé par Bouhaddani *et al.* (2018).

Nous comparerons la régression *NIPALS-PLS* avec la régression *PLS* probabiliste dans les deux situations de données complètes et données incomplètes. Pour voir la robustesse de ces modèles, nous conserverons toujours un certain nombre de paramètres : les dimensions des jeux de données, le nombre vrai de composantes, les critères, les proportions de données manquantes et les hypothèses de données manquantes pour déterminer le nombre de composantes.

10.1 Introduction

Plusieurs perspectives peuvent être envisagées. Nous avons présenté la perspective du travail sur laquelle des travaux sont déjà en cours. L'algorithme *NIPALS* peut se révéler inadapté pour un jeu de données dans lequel il y a beaucoup de données manquantes. En effet, il ignore les valeurs manquantes lors du calcul des matrices de composantes du modèle de la régression *PLS*. Cet algorithme peut donc ne pas fonctionner correctement pour construire un modèle de régression *PLS* à partir d'un jeu de données où il y a beaucoup de données manquantes.

L'idée de l'ACPP a été développée en 1999 par Tipping et Bishop (Tipping and Bishop, 1999). Cette méthode a été utilisée avec succès pour estimer les données manquantes et de nombreux développements actuels des méthodes statistiques utilisent également ce concept probabiliste. Ainsi, la régression *PLS* probabiliste établit une relation entre une matrice de prédicteurs **X** et une variable à prédire **Y** sous la forme d'un modèle combinant l'ACPP et un modèle de régression

linéaire, ou non-linéaire. Ce modèle a été proposé par Li *et al.* (2011), Li *et al.* (2015), Zheng *et al.* (2016) et Bouhaddani *et al.* (2018). Ainsi, une des perspectives du travail présentée ici est de développer cette idée au cas de la détermination du nombre de composantes d'un modèle de régression *PLS*, sur des données complètes puis sur des données incomplètes.

10.2 Présentation du modèle de l'analyse en composantes principales probabiliste

L'ACPP dérive d'un modèle à variables cachées, avec les hypothèses d'un bruit isotrope et d'un a priori gaussien. En utilisant la notation de Li *et al.* (2011), $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, le vecteur observé $\mathbf{x}$ de dimension p est généré à partir du vecteur caché $\mathbf{t}$ de dimension H suivant l'expression :

$$\mathbf{x} = \mathbf{P}\mathbf{t} + \mu_x + \boldsymbol{\epsilon} \tag{10.1}$$

où $\mathbf{P}$ est une matrice de dimension $p \times H$, μ est un vecteur de la moyenne des données et $\boldsymbol{\epsilon}$ est un bruit gaussien de moyenne nulle et de matrice de covariance $\sigma_x^2\mathbb{I}$, $\mathbb{I}$ étant la matrice identité $p \times p$.

La distribution de $\mathbf{x}$ connaissant $\mathbf{t}$ suit une densité gaussienne :

$$\begin{aligned}\mathbb{P}(\mathbf{x}|\mathbf{t}) &= (2\pi\sigma_x^2)^{-\frac{p}{2}} exp\left\{-\frac{1}{2\sigma_x^2}|\mathbf{x} - \mathbf{P}\mathbf{t} - \mu_x|^2\right\} \\ &= \mathcal{N}(\mathbf{x}|\mathbf{P}\mathbf{t} + \boldsymbol{\mu}_x, \sigma_x^2\mathbb{I})\end{aligned} \tag{10.2}$$

Un a priori gaussien est choisi pour $\mathbf{t}$:

$$\begin{aligned}\mathbb{P}(\mathbf{t}) &= (2\pi)^{-\frac{H}{2}} exp\left\{-\frac{1}{2}\mathbf{t}'\mathbf{t}\right\} \\ &= \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbb{I})\end{aligned} \tag{10.3}$$

ce qui donne la distribution gaussienne du vecteur $\mathbf{x}$:

$$\mathbb{P}(\mathbf{x}) = (2\pi)^{-\frac{p}{2}} |\mathbf{A}|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_x)'\mathbf{A}^{-1}(\mathbf{x} - \mu_x)\right\} \tag{10.4}$$

avec $\mathbf{A} = \sigma_x^2\mathbb{I} + \mathbf{P}\mathbf{P}'$ une matrice $p \times p$.

La distribution a posteriori est donnée par :

$$\mathbb{P}(\mathbf{t}|\mathbf{x}) = \left(\frac{1}{2\pi\sigma_x^2}\right)^{\frac{H}{2}} |\mathbf{M}|^{\frac{1}{2}} exp\left[\frac{-1}{2}\left\{\mathbf{t} - \mathbf{M}^{-1}\mathbf{P}'(\mathbf{x} - \mu_x)\right\}'\left(\sigma_x^{-2}\mathbf{M}\right)\left\{\mathbf{t} - \mathbf{M}^{-1}\mathbf{P}'(\mathbf{x} - \mu_x)\right\}\right] \tag{10.5}$$

avec $\mathbf{M} = \sigma^2\mathbb{I} + \mathbf{P}'\mathbf{P}$ une matrice de dimension $H \times H$.

La log-vraisemblance des données par rapport au modèle est :

$$\begin{aligned} L &= \sum_{i=1}^{n} \ln\left(\mathbb{P}(x_i)\right) \\ &= \frac{n}{2}\left\{p\ln(2\pi) + \ln|\mathbf{A}| + tr(\mathbf{A}^{-1}\mathbf{S}_x)\right\} \end{aligned} \tag{10.6}$$

où $\mathbf{S}_x = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mu_x)(\mathbf{x}_i - \mu_x)\prime$ est la matrice de covariance des données observées.

10.3 Présentation du modèle de la régression *PLS* probabiliste

Le principe général de la régression *PLS* probabiliste est le suivant : soit $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{pxn}$ représente p lignes pour chaque variable explicative et $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n) \in \mathbb{R}^{qxn}$ représente q lignes pour chaque variable réponse. Les matrices $\mathbf{X}$ et $\mathbf{Y}$ peuvent être caractérisées comme une combinaison linéaire de composantes $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_n) \in \mathbb{R}^{Hxn}$. Le modèle de *PLS* probabiliste s'écrit donc :

$$\begin{cases} \mathbf{x} = \mathbf{P}\mathbf{t} + \mu_x + \epsilon \\ \mathbf{y} = \mathbf{C}\mathbf{t} + \mu_y + \varepsilon \end{cases} \tag{10.7}$$

où $\mathbf{P}$ et $\mathbf{C}$ sont les coefficients des composantes pour $\mathbf{X}$ et $\mathbf{Y}$. ϵ_i et ε_i sont les vecteurs des résidus qui sont la différence entre l'approximation et les données observées. μ_x et μ_y représentent les vecteurs moyens de $\mathbf{X}$ et $\mathbf{y}$. En désignant $\mathcal{N}$ comme une distribution gaussienne, $\epsilon \sim \mathcal{N}(0, \sigma_x^2\mathbb{I})$ et $\varepsilon \sim \mathcal{N}(0, \sigma_y^2\mathbb{I})$ spécifient les bruits gaussiens du modèle avec des moyennes nulles et des covariances de $\sigma_x^2\mathbb{I}$ et $\sigma_y^2\mathbb{I}$, respectivement. Sur la base des modèles de bruit gaussien des vecteurs ϵ et ε, les probabilités conditionnelles de $\mathbf{x}$ et $\mathbf{y}$ étant données $\mathbf{t}$ sont écrites comme suit (Li *et al.*, 2011) :

$$\mathbb{P}(\mathbf{x}|\mathbf{t}) = \mathcal{N}(\mathbf{x}|\mathbf{P}\mathbf{t} + \mu_x, \sigma_x^2\mathbb{I}), \tag{10.8}$$

et

$$\mathbb{P}(\mathbf{y}|\mathbf{t}) = \mathcal{N}(\mathbf{y}|\mathbf{C}\mathbf{t} + \mu_y, \sigma_y^2\mathbb{I}). \tag{10.9}$$

La distribution *a priori* gaussienne de $\mathbf{t}$ est donnée par :

$$\mathbb{P}(\mathbf{t}) = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbb{I}). \tag{10.10}$$

Par une règle bayésienne, nous pouvons avoir les distributions *a posteriori* de $\mathbf{t}$:

$$\mathbb{P}(\mathbf{t}|\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x}|\mathbf{t})\ \mathbb{P}(\mathbf{t})}{\mathbb{P}(\mathbf{x})} = \mathcal{N}(\mathbf{t}|\mathbf{M}^{-1}\mathbf{P}'(\mathbf{x} - \mu_x)^2, \sigma_x^2\mathbf{M}^{-1}) \tag{10.11}$$

avec $\mathbf{M} = \sigma_x^2\mathbb{I} + \mathbf{P}'\mathbf{P}$

Les distributions marginales de $\mathbf{x}$ et $\mathbf{y}$ s'obtiennent alors facilement par intégration ; elles sont

gaussiennes :

$$\mathbb{P}(\mathbf{x}) = \int \mathbb{P}(\mathbf{x}|\mathbf{t})\,\mathbb{P}(\mathbf{t})\,d\mathbf{t} = \mathcal{N}(\mathbf{x}|\mu_x, \mathbf{PP}' + \sigma_x^2\mathbb{I}), \tag{10.12}$$

$$\mathbb{P}(\mathbf{y}) = \int \mathbb{P}(\mathbf{y}|\mathbf{t})\,\mathbb{P}(\mathbf{t})\,d\mathbf{t} = \mathcal{N}(\mathbf{y}|\mu_y, \mathbf{CC}' + \sigma_y^2\mathbb{I}), \tag{10.13}$$

La distribution conjointe de $\mathbf{Y}$ et $\mathbf{X}$ à partir de (10.8) et (10.9) est :

$$\mathbb{P}(\mathbf{x}, \mathbf{y}|\mathbf{t}, \Theta) = \mathcal{N}(\mathbf{x}, \mathbf{y}|\mu_{xy}, \boldsymbol{S}_{xy}) \tag{10.14}$$

où $\mathbf{u}_{xy}$ est la moyenne et $\boldsymbol{S}_{xy}$ est la covariance et $\Theta = \left\{\mathbf{P}, \mathbf{C}, \mu_x, \mu_y, \sigma_x^2, \sigma_y^2\right\}$ représentent l'ensemble des paramètres du modèle *PLS* probabiliste.

$$\begin{aligned} \mathbf{u}_{xy} &= \begin{pmatrix} \mathbf{Pt} + \mu_x \\ \mathbf{Ct} + \mu_y \end{pmatrix} \\ \boldsymbol{S}_{xy} &= \begin{pmatrix} \sigma_x^2\mathbb{I} & \mathbf{0} \\ \mathbf{0} & \sigma_y^2\mathbb{I} \end{pmatrix}, \end{aligned} \tag{10.15}$$

En utilisant les équations 10.14 et 10.10, la distribution *a posteriori* de $\mathbf{t}$ sur $\mathbf{x}$ and $\mathbf{y}$ par le théorème de Bayes s'écrit ainsi :

$$\mathbb{P}(\mathbf{t}|\mathbf{x}, \mathbf{y}, \Theta) = \mathcal{N}(\mathbf{t}|\boldsymbol{\mu}_t, \mathbf{S}_t) \tag{10.16}$$

avec

$$\begin{aligned} \boldsymbol{\mu}_t &= \boldsymbol{S}_t \boldsymbol{\Lambda}' \boldsymbol{S}_{xy}^{-1} (\mathbf{z} - \boldsymbol{\mu}_z) \\ \boldsymbol{S}_t &= \left(\mathbb{I} + \boldsymbol{\Lambda}' \boldsymbol{S}_{xy}^{-1} \boldsymbol{\Lambda}\right)^{-1}, \end{aligned} \tag{10.17}$$

où $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$, $\boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{P} \\ \mathbf{C} \end{bmatrix}$ et $\boldsymbol{\mu}_z = \begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}$.

Étant donné que les composantes et les paramètres du modèle sont inconnus dans la fonction de vraisemblance de l'équation 10.14, il est difficile d'obtenir simultanément des solutions pour les composantes et les valeurs des paramètres à partir du modèle *PLS* probabiliste. Par conséquent, l'algorithme *EM* peut être utilisé pour estimer de manière itérative les composantes et les paramètres du modèle à partir de données complètes (Li *et al.*, 2011).

10.4 Cadre général de l'algorithme *EM* pour la régression *PLS* probabiliste

Les paramètres $\Theta = \left\{\mathbf{P}, \mathbf{C}, \mu_x, \sigma_x, \mu_y, \sigma_y\right\}$ sont inconnus même si nous avons déjà construit le modèle sur les variables $\mathbf{x}$, $\mathbf{y}$ et $\mathbf{t}$. En observant les échantillons $\mathbf{X}$ et $\mathbf{Y}$, nous pouvons obtenir la fonction de log-vraisemblance :

$$L = \ln \mathbb{P}(\mathbf{Y}, \mathbf{X}; \Theta) = \sum_{i=1}^{n} \ln \mathbb{P}(\mathbf{y}_i, \mathbf{x}_i; \Theta) \tag{10.18}$$

Afin d'optimiser les paramètres pour maximiser la fonction de log-vraisemblance, nous avons besoin d'obtenir la dérivée de la fonction de log-vraisemblance et de trouver la valeur pour laquelle elle s'annule. Nous prenons d'abord l'algorithme *EM* pour initialiser tous les paramètres Θ. Avec l'équation 10.16, la distribution des composantes peut être calculée et la fonction log-vraisemblance est écrite ainsi (Li *et al.*, 2011) :

$$\begin{aligned} L = -\sum_{i=1}^{n} \Bigg\{ & \frac{p}{2}\ln 2\pi\sigma_x^2 - \frac{1}{\sigma_x^2}\mathbb{E}[\mathbf{t}_i]'\mathbf{P}'(\mathbf{x}_i-\boldsymbol{\mu}_x) + \frac{1}{2\sigma_x^2}\|\mathbf{x}_i-\boldsymbol{\mu}_x\|^2 + \frac{1}{2\sigma_x^2}\mathrm{tr}\left(\mathbb{E}\left[\mathbf{t}_i\mathbf{t}_i'\right]\ \mathbf{P}'\mathbf{P}\right) \\ & + \frac{q}{2}\ln(2\pi\sigma_y^2) - \frac{1}{\sigma_y^2}\mathbb{E}[\mathbf{t}_i]'\mathbf{C}'(\mathbf{y}_i-\boldsymbol{\mu}_y) + \frac{1}{2\sigma_y^2}\left\|\mathbf{y}_i-\boldsymbol{\mu}_y\right\|^2 + \frac{1}{2\sigma_y^2}\mathrm{tr}\left(\mathbb{E}\left[\mathbf{t}_i\mathbf{t}_i'\right]\ \mathbf{C}'\mathbf{C}\right) \Bigg\} \end{aligned} \tag{10.19}$$

Dans l'algorithme *EM*, l'étape *E* et l'étape *M* sont conduites de manière itérative jusqu'à ce que les paramètres du modèle convergent. Après avoir obtenu ces paramètres, la relation entre $\mathbf{x}$ et $\mathbf{y}$ avec la distribution marginale $\mathbf{t}$ peut être modélisée par :

$$\mathbb{P}(\mathbf{y}|\mathbf{x}) = \int \mathbb{P}(\mathbf{y}|\mathbf{t})\,\mathbb{P}(\mathbf{t}|\mathbf{x})\,d\mathbf{t}, \tag{10.20}$$

En combinant les équations 10.9 et 10.11, l'équation10.20 peut s'écrire :

$$\mathbb{P}(\mathbf{y}|\mathbf{x},\mathbf{C},\mathbf{P}) = \mathcal{N}(\mathbf{y}|\boldsymbol{\mu},\mathbf{S}) \tag{10.21}$$

avec :

$$\begin{aligned} \boldsymbol{\mu} &= \mathbf{C}(\mathbf{P}'\mathbf{P}+\sigma_x^2\mathbb{I})^{-1}\mathbf{P}'(\mathbf{x}-\boldsymbol{\mu}_x)+\boldsymbol{\mu}_y \\ \mathbf{S} &= \mathbf{C}\sigma_x^2(\mathbf{P}'\mathbf{P}+\sigma_x^2\mathbb{I})^{-1}\mathbf{C}'+\sigma_y^2\mathbb{I} \end{aligned} \tag{10.22}$$

Donc, les coefficients de la régression *PLS* probabiliste sont obtenus comme suit :

$$\beta = \mathbf{C}\left(\mathbf{P}'\mathbf{P}+\sigma_x^2\mathbb{I}\right)^{-1}\mathbf{P}' \tag{10.23}$$

Les étapes de l'algorithme *EM* pour la régression *PLS* probabiliste peuvent se voir dans l'algorithme ci-dessous (Li *et al.*, 2011) :

1. Les matrices de données (explicatives ou réponses) sont supposées centrées et réduites.
2. Initialisation : $\sigma_x^2 = 1$, $\sigma_y^2 = 1$, $\mathbf{P}$ = la première colonne H de la matrice $\mathbf{X}$ et $\mathbf{C}$ = la première colonne H de la matrice $\mathbf{Y}$
3. Répéter jusqu'à convergence de $\boldsymbol{\mu}_t$
   (a) Calculer $\boldsymbol{\mu}_t$ et $\boldsymbol{S}_t$ comme dans l'équation 10.17
   (b) Calculer $\mathbf{P}$, $\mathbf{C}$, σ_x^2 et σ_y^2 comme dans les équations 10.26 à 10.29

10.5 Algorithme *EM* pour la régression *PLS* probabiliste avec des données manquantes

La régression *PLS* probabiliste proposée vise à gérer les valeurs manquantes en tenant compte de l'incertitude avec une stratégie probabiliste. Afin d'estimer les composantes et les paramètres d'un modèle par algorithme *EM*, les données manquantes aléatoires sont initialement imputées par des valeurs initiales et les valeurs des paramètres du modèle sont fixées par une estimation initiale. Ensuite, les étapes *E* et *M* de l'algorithme *EM* peuvent être effectuées de manière itérative. Nous proposerons l'initialisation des données manquantes dans une matrice $\mathbf{X}$ en utilisant une imputation par *MICE*.

THÉORÈME 1 *Mise à jour des paramètres du modèle et des données manquantes en remplissant la relation suivante :*

$$\boldsymbol{\mu}_x^{k+1} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}_i^k - \boldsymbol{P}^k\mathbb{E}\left[\boldsymbol{t}_i^k\right]\right) \tag{10.24}$$

$$\boldsymbol{\mu}_y^{k+1} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{y}_i^k - \boldsymbol{C}^k\mathbb{E}\left[\boldsymbol{t}_i^k\right]\right) \tag{10.25}$$

$$\boldsymbol{P}^{k+1} = \left[\sum_{i=1}^{n}\left(\boldsymbol{x}_i^k - \boldsymbol{\mu}_x^k\right)\mathbb{E}\left[\boldsymbol{t}_i^k\right]'\right]\left[\sum_{i=1}^{n}\mathbb{E}\left[\boldsymbol{t}_i^k(\boldsymbol{t}_i^k)'\right]\right]^{-1} \tag{10.26}$$

$$\boldsymbol{C}^{k+1} = \left[\sum_{i=1}^{n}\left(\boldsymbol{y}_i^k - \boldsymbol{\mu}_y^k\right)\mathbb{E}\left[\boldsymbol{t}_i^k\right]'\right]\left[\sum_{i=1}^{n}\mathbb{E}\left[\boldsymbol{t}_i^k\boldsymbol{t}_i^{k\prime}\right]\right]^{-1} \tag{10.27}$$

$$\sigma_x^{2,k+1} = \frac{1}{np}\sum_{i=1}^{n}\left\{\left\|\boldsymbol{x}_i^k - \boldsymbol{\mu}_x^k\right\|^2 - 2\mathbb{E}\left[\boldsymbol{t}_i^k\right]'\boldsymbol{P}^{k\prime}\left(\boldsymbol{x}_i^k - \boldsymbol{\mu}_x^k\right) + tr\left(\mathbb{E}\left[\boldsymbol{t}_i^k\boldsymbol{t}_i^{k\prime}\right]\boldsymbol{P}^{k\prime}\boldsymbol{P}^k\right)\right\} \tag{10.28}$$

$$\sigma_y^{2,k+1} = \frac{1}{nq}\sum_{i=1}^{n}\left\{\left\|\boldsymbol{y}_i^k - \boldsymbol{\mu}_y^k\right\|^2 - 2\mathbb{E}\left[\boldsymbol{t}_i^k\right]'\boldsymbol{C}^{k\prime}\left(\boldsymbol{y}_i^k - \boldsymbol{\mu}_y^k\right) + tr\left(\mathbb{E}\left[\boldsymbol{t}_i^k\boldsymbol{t}_i^{k\prime}\right]\boldsymbol{C}^{k\prime}\boldsymbol{C}^k\right)\right\} \tag{10.29}$$

$$\boldsymbol{x}_i^{k,m} = \boldsymbol{P}^{k,m}\mathbb{E}\left[\boldsymbol{t}_i^k\right] + \boldsymbol{\mu}_x^{k,m} \tag{10.30}$$

$$\boldsymbol{y}_i^{k,m} = \boldsymbol{C}^{k,m}\mathbb{E}\left[\boldsymbol{t}_i^k\right] + \boldsymbol{\mu}_y^{k,m} \tag{10.31}$$

où k dans l'exposant est le nombre d'itérations, $\mathbb{E}[.]$ *est l'espérance,* $tr()$ *est la trace de la matrice et* $\boldsymbol{\mu}_x^{k,m}$ *et* $\boldsymbol{\mu}_y^{k,m}$ *sont les sous-vecteurs de* $\boldsymbol{\mu}_x^k$ *et*$\boldsymbol{\mu}_y^k$ *correspondant respectivement à* $\boldsymbol{x}_i^m$ *et* $\boldsymbol{y}_i^m$. $\boldsymbol{x}_i^m$ *et* $\boldsymbol{y}_i^m$ *sont les vecteurs de données manquantes. Dans chaque itération, le premier et le deuxième ordre statistique sont calculés par* $\mathbb{E}[\boldsymbol{t}_i] = \boldsymbol{S}_t\boldsymbol{\Lambda}'\boldsymbol{S}_{xy}^{-1}(\boldsymbol{z} - \boldsymbol{\mu}_z)$ *et* $\mathbb{E}[\boldsymbol{t}_i\boldsymbol{t}_i'] = \boldsymbol{S}_t + \mathbb{E}[\boldsymbol{t}_i]\,\mathbb{E}[\boldsymbol{t}_i]'$, *respectivement.*

Dans l'algorithme *EM*, l'étape *E*, l'étape *M* et l'étape de ré-estimation sont effectuées de manière itérative jusqu'à ce que les paramètres du modèle convergent. Ces étapes sont résumées sur la Figure 10.1.

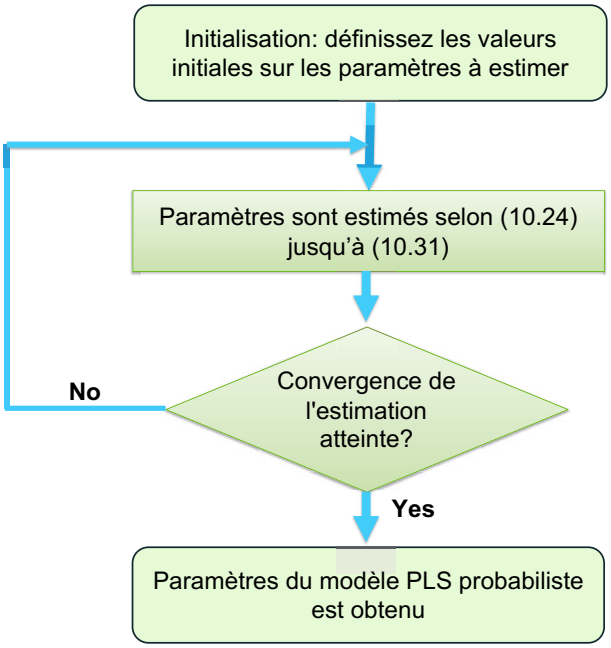


Figure 10.1 – Procédure de la modélisation *PLS* probabiliste sur les données manquantes.

10.6 Processus de simulations

Nous proposons le processus de simulation comme ci-dessous :

1. Les données sont simulées sur la base de Li *et al.* (2002).
   Le nombre vrai de composantes (t^*) est égal à 2, 3, 4, 5 et 6.
   Le nombre d'observations (n) et le nombre de variables (p) respectent dans les combinaisons $n = 100$ et $p = 20$.

2. Les données manquantes sont créées sous l'hypothèse d'un mécanisme *MCAR* et *MAR* avec une proportion de données manquantes (d) allant de 5% à 50% par pas de 5%.

3. Les valeurs manquantes sont initialisées par les valeurs de la méthode *MICE* avec un nombre d'imputation (m) égal à cinq.
   Le nombre de composantes est ensuite estimé en utilisant la régression *PLS* probabiliste à l'aide d'un *package* `PO2PLS : Probabilistic Two-Way Orthogonal Partial Least Squares` (Bouhaddani, 2017).
   Le nombre de composantes est choisi à l'aide d'une validation croisée avec le critère Q^2, soit Q^2-*LOO* soit Q^2-10-*Fold*, et des critères *AIC* et *BIC*.
   Le nombre de composantes retenues est le mode du nombre de composantes obtenus par les critères sur l'ensemble des imputation.

4. Pour chaque combinaison du nombre vrai de composantes, de la proportion de données manquantes, du mécanisme supposé d'apparition des données manquantes, 100 réplications ont été tirés.

À partir de ces 100 simulations, nous comptons finalement le nombre de composantes retenues qui correspondent à t^* déterminé. Ce processus de simulations est illustré sur la Figure 10.2.

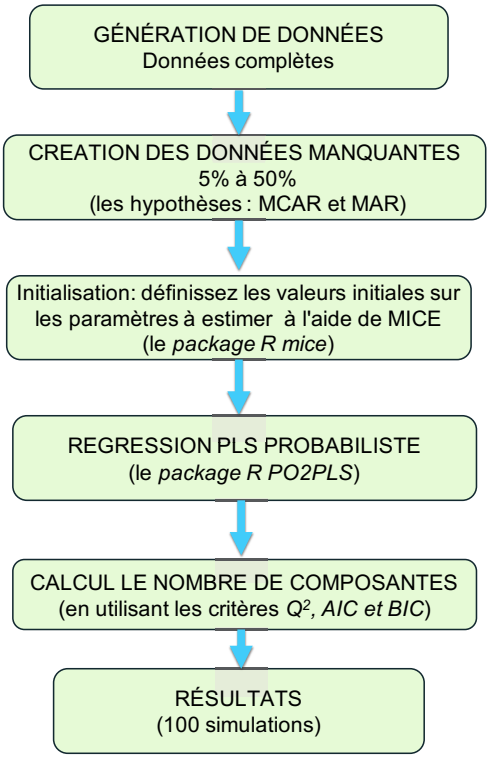


Figure 10.2 – Procédure de la simulation.

10.7 Résultats obtenus

10.7.1 Données complètes

Le Tableau 10.1 montre la fréquence du nombre de composantes retenues dans les cas de données complètes. Dans ce tableau présente le nombre de composantes retenues qui est égal au nombre vrai de composantes (2, 3, 4, 5 et 6 composantes) sur 100 simulations.

De manière générale, les résultats du nombre de composantes retenues selon les critères Q^2-*LOO*, Q^2-10-*Fold*, *AIC* et *BIC* montrent qu'ils ont de mauvaises performances. Dans la grande majorité des situations, les performances des critères Q^2-*LOO* et Q^2-10-*Fold* tendent à sélectionner un nombre de composantes plus petit que le nombre vrai de composantes. Ils ont tendance à choisir un nombre de composantes qui est égal à 1 ou 2. En revanche, les critères *AIC* et *BIC* tendent parfois à choisir trop de composantes et parfois moins de composantes. Leurs performances dépendent du nombre vrai de composantes (voir lignes 3 et 4 du Tableau 10.1 pour chaque t^*).

10.7.2 Données incomplètes

Nous résumons les résultats des simulations de toutes les combinaisons de critères en fonction des mécanismes et de la proportion de données manquantes à l'Annexe E dans les Tableaux E.1 à E.10. Les résultats de ces tableaux présentent le nombre de composantes retenues qui est égal

Table 10.1 – Nombre de composantes retenues pour une matrice de dimension $n = 100$ et $p = 20$ selon t^* et selon les critères, sur 100 simulations.

t^*	Critères	Nombre de composantes retenues								Moyenne du nombre de composantes
		1	2	3	4	5	6	7	8	
2	Q^2-*LOO*	0	1	99	0	0	0	0	0	2.99
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	1	0	0	0	99	0	6.96
	BIC	0	0	1	0	0	0	99	0	6.96
3	Q^2-*LOO*	50	50	0	0	0	0	0	0	1.50
	Q^2-10-*Fold*	50	50	0	0	0	0	0	0	1.50
	AIC	0	0	0	48	0	0	2	50	6.06
	BIC	0	0	0	48	0	0	2	50	6.06
4	Q^2-*LOO*	35	3	60	1	1	0	0	0	2.30
	Q^2-10-*Fold*	65	34	0	1	0	0	0	0	1.37
	AIC	0	64	1	30	1	2	1	1	2.83
	BIC	0	64	1	30	1	2	1	1	2.83
5	Q^2-*LOO*	3	93	2	1	1	0	0	0	2.04
	Q^2-10-*Fold*	3	95	0	2	0	0	0	0	2.01
	AIC	0	1	5	2	2	1	88	1	6.65
	BIC	0	1	5	2	2	1	88	1	6.65
6	Q^2-*LOO*	76	24	0	0	0	0	0	0	1.24
	Q^2-10-*Fold*	49	27	0	0	24	0	0	0	2.23
	AIC	0	0	26	0	24	25	0	25	5.48
	BIC	0	0	26	0	24	25	0	25	5.48

à 2, 3, 4, 5 ou 6 sur 100 simulations pour lesquelles la proportion de données manquantes (d) allant de 5% à 50% par pas de 5% et t^*, aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR*.

Les performances de tous les critères sont globalement mauvaises pour déterminer le nombre vrai de composantes, quelle que soit la proportion de données manquantes et l'hypothèse sur le mécanisme des données manquantes. Dans la plupart des types de dimensions de données, le nombre de composantes retenues avec les critères Q^2-*LOO* et Q^2-10-*Fold* a tendance à être inférieur au nombre vrai de composantes. Le nombre de composantes retenues avec ces critères est le plus proche du nombre de composantes qui est égal à 1 ou 2 composantes, soit sur l'hypothèse MCAR soit l'hypothèse MAR, et ceci avec n'importe quelle proportion de donnes manquantes.

Comme pour les données complètes, les critères *AIC* et *BIC* tendent parfois à choisir un nombre trop grand de composantes ou un nombre petit de composantes. Leurs performances dépendent du nombre vrai de composantes et de la proportion de données manquantes (voir lignes 3 et 4 des tableaux dans l'Annexe E pour chaque t^*).

10.8 Conclusion des simulations pour la régression *PLS* probabiliste

Ce travail est en cours (et c'est pourquoi il est placé dans la partie Perspectives et Conclusions) et peu de résultats ont été obtenus pour l'instant. Un élément qui semble très important est que la détermination du nombre de composantes dans la régression *PLS* probabiliste ne donne pas de résultats satisfaisants. En fait, les quatre critères utilisés pour déterminer le nombre vrai de composantes ont de moins bonnes performances tant sur données complètes que sur données incomplètes.

Cette situation peut être due à une modification des réponses multivariées aux réponses univariées dans le *package* `PO2PLS`. Ce *package* est fondamentalement utilisé pour des réponses multivariées. Ainsi, une autre perspective intéressante serait le développement de la réponse univariée à la réponse multivariée pour déterminer le nombre de composantes dans la régression *PLS* probabiliste sur les données complètes et sur les données incomplètes.

Chapitre 11

Conclusion

11.1 Conclusion

Dans la recherche et le développement, que cela soit dans le domaine industriel ou académique, les données manquantes sont un réel problème pour le praticien. D'une part, elles sont très souvent présentes et semblent presque inévitables dans des jeux de données de grandes dimensions, les domaines de la biologie et de la santé n'échappant pas à cette règle. D'autre part, elles peuvent provoquer des problèmes d'estimation, comme par exemple, provoquer des biais dans les estimateurs qui impliquent alors dans l'analyse du jeu de données et dans l'interprétation, des résultats faux.

Plusieurs approches statistiques ont été développées pour traiter les données manquantes. Nous pouvons citer à titre d'exemple les techniques d'imputation qui consistent à remplacer les données manquantes par une valeur générée au cours du processus d'imputation. L'objectif de l'imputation est d'utiliser des relations connues qui peuvent être identifiées dans les observations non manquantes du jeu de données pour estimer la valeur des données manquantes.

La régression *PLS* (*Partial Least Squares*) est une méthode statistique de modélisation, appartenant aux méthodes de régression. Cette méthode de régression repose essentiellement sur un algorithme bien connu, le *NIPALS*. L'algorithme *NIPALS* a l'avantage de pouvoir estimer les composantes même lorsque les données sont incomplètes, dans la mesure où chaque composante est estimée à partir des seules données complètes, de manière itérative sur chaque dimension du jeu de données et ceci, sans devoir recourir à l'imputation. De cette combinaison du principe de la régression et de l'algorithme *NIPALS* est née la régression *NIPALS-PLS*. Elle se révèle comme une méthode d'analyse des données qui permet de traiter le cas particulier des données de grandes dimensions, éventuellement fortement corrélées et potentiellement incomplètes. La détermination du nombre de composantes construites lors de la régression *PLS* ne tient pas compte ni du type de manquant ni de la proportion de données manquantes dans le jeu de données. Pourtant il s'agit d'un point essentiel pour établir des modèles de régression fiables ainsi que pour sélectionner correctement des prédicteurs. La détermination du nombre de composantes d'un modèle de régression *PLS* sur données incomplètes est le sujet principal de cette thèse.

Au cours de ces trois années de thèse, nous avons conduit une évaluation de la régression

NIPALS-PLS en cas de données manquantes. Dans la détermination du nombre de composantes, plusieurs critères ont été étudiés, notamment les critères de la validation croisée et les critères d'information, avec des simulations. Les critères de validation croisée sont : le Q^2 avec *Leave-One-Out* (Q^2-*LOO*) et le Q^2 avec 10-*Fold* (Q^2-10-*Fold*). Les critères d'information sont d'une part le critère *AIC* et d'autre part le critère *BIC*. Ces derniers critères ont été évalués de manière naïve ou en faisant appel à la notion de degrés de liberté (*DoF*) de la régression *PLS*, telle qu'elle a été introduite par Krämer and Sugiyama (2011). Ainsi, nous avons étudié les performances de six critères : Q^2-*LOO*, Q^2-10-*Fold*, *AIC*, *AIC-DoF*, *BIC* et *BIC-DoF*.

Nous avons également comparé les performances de ces critères sur un jeu de données incomplet et sur un jeu de données imputé en utilisant trois méthodes d'imputation : l'imputation *MICE* (Van Buuren and Groothuis-Oudshoorn, 2011), l'imputation *KNN* (Kowarik and Templ, 2016) et l'imputation par *SVD* (Perry, 2015). Nous avons testé les critères avec des proportions différentes de données manquantes (de 5% à 50%) sous différentes hypothèses de type de données manquantes (*MCAR* et *MAR*). Différentes dimensions des jeux de données ont également été testés avec deux grands cas de figure : $n > p$ (matrice verticale) et $n < p$ (matrice horizontale). Des simulations ont été réalisées pour évaluer les performances des différents critères ; ces simulations utilisent la méthode décrite par Li *et al.* (2002).

Les critères Q^2-*LOO* et Q^2-10-*Fold* peuvent être considérés comme le meilleur choix de critères pouvant être utilisés pour déterminer le nombre de composantes dans la construction d'un modèle de régression *PLS*. Ces critères donnent les meilleurs résultats, tant sur les données complètes que sur les données incomplètes. Les critères *AIC*, *AIC-DoF*, *BIC* et *BIC-DoF* tendent à sélectionner plus de composantes qu'il n'en faut. En effet, le nombre de composantes retenues avec ces critères est généralement trop grand.

Les résultats de la comparaison de quatre méthodes sont très intéressants. Les résultats montrent que le nombre vrai de composantes et les dimensions des données affectent les résultats des méthodes utilisées. En général, lorsque le nombre vrai de composantes est égal à deux, la méthode *NIPALS-PLS* avec Q^2-*LOO* fournit une performance satisfaisante avec n'importe quelle dimension de jeux de données sous l'hypothèse *MCAR*. En comparaison, lorsque le nombre vrai de composantes est égal à quatre ou six, la méthode *MICE* avec les critères Q^2-*LOO* et Q^2-10-*Fold* a de bonnes performances aussi bien sous l'hypothèse *MCAR* que sous l'hypothèse *MAR* lorsque les dimensions de données sont verticales.

Enfin, ce travail est intéressant, car la plupart des propriétés observées et étudiées (critères, différentes dimensions pour des jeux de données, différents nombres vrais de composantes, proportions différentes de données manquantes, différentes hypothèses de données manquantes (*MCAR* et *MAR*) et différentes méthodes d'imputation) ont été analysés globalement. Ce résultat est lié à la robustesse de la régression *PLS* dans le cas de données incomplètes. Nous espérons à partir de ces résultats que les praticiens pourront considérer les critères et la proportion de données manquantes pour produire de meilleurs modèles.

11.2 Perspectives

Après ce travail, plusieurs perspectives peuvent être envisagées. Tout d'abord, au chapitre 10, nous avons présenté un aperçu de travaux qui sont déjà en cours. En effet, nous avons proposé

une méthode pour estimer la détermination du nombre de composantes dans la régression *PLS* en examinant la relation entre une matrice de variables explicatives et une variable réponse dans les cas de données incomplètes.

Les résultats préliminaires obtenus ont indiqué que la régression *PLS* probabiliste n'ont pas de bonnes performances, ni dans le critère Q^2, ni dans le critère *AIC*, ni dans le critère *BIC* sur la réponse univariée. Comme expliqué précédemment, nous pouvons développer la détermination du nombre de composants de la régression *PLS* probabiliste sur la réponse multivariée.

D'autres pistes sont envisageables pour développer la régression *NIPALS-PLS* en particulier l'analyse de la capacité prédictive du modèle et de la sélection des variables. Elles semblent être une perspective intéressante. Le choix de variables sera rendu possible en comparant la méthode des arbres *CART* (en anglais *Classification And Regression Trees*) qui a été introduit pour la première fois par Breiman *et al.* (1984b).

Une autre perspective est la parallélisation du travail à l'aide des cartes graphiques. Le modèle de régression *PLS* nécessite un long temps de calcul, en raison du nombre élevé d'itérations requises dans l'algorithme de la régression. Il est donc important de pouvoir optimiser les temps de calculs, notamment lorsque des simulations doivent être effectuées pour tester telle ou telle propriété du modèle. Une programmation de méthode *PLS* sur des cartes graphiques (en anglais *Graphics Processing Unit - GPU*) permettrait d'accélérer notablement les temps de calculs (Srinivasan *et al.*, 2010).

Le calcul par le *GPU* permet de paralléliser les tâches et d'offrir un maximum de performances dans de nombreuses applications : le *GPU* accélère les portions de code les plus lourdes en ressources de calcul, le reste de l'application restant affecté à l'utilisation du processeur central (*Central Processing Unit – CPU*). Les applications des utilisateurs s'exécutent ainsi bien plus rapidement.

Cinquième partie

Annexes

Annexe A

Schéma de calcul des degrés de liberté de la régression *PLS*

Dans le cadre de la régression *PLS*, le schéma de calcul des degrés de liberté est présenté à la figure A.1.

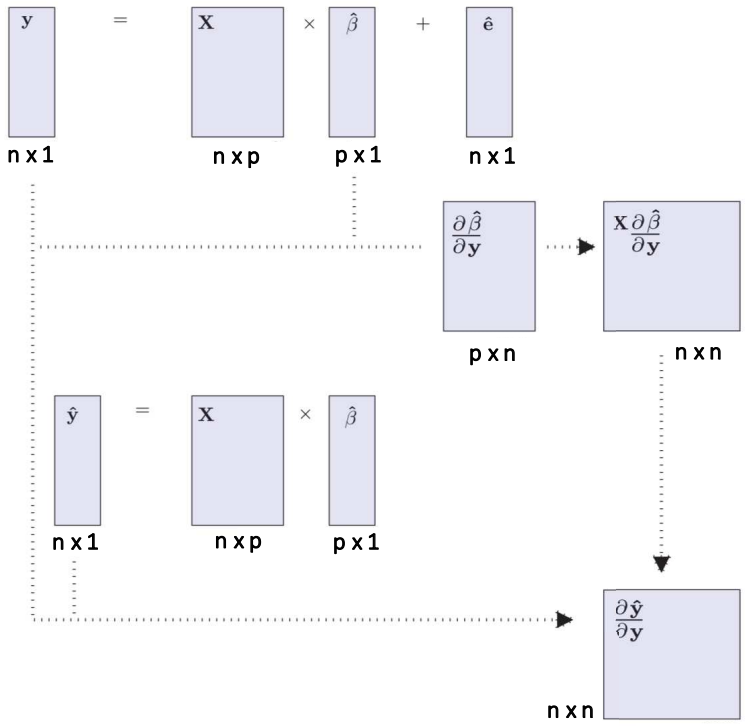


Figure A.1 – Schéma du calcul des degrés de liberté dans la régression *PLS*.

Annexe B

Évaluation du nombre de composantes retenues lors des simulations de la régression *NIPALS-PLS*

Dans cette annexe, nous fournissons les résultats des simulations selon les critères, les méthodes et le nombre vrai de composantes en fonction des mécanismes et de la proportion de données manquantes sur les matrices de forme verticale et sur les matrices de forme horizontale.

B.1 Données à matrice verticale

B.1.1 Nombre vrai de composantes = 2

Q^2-LOO

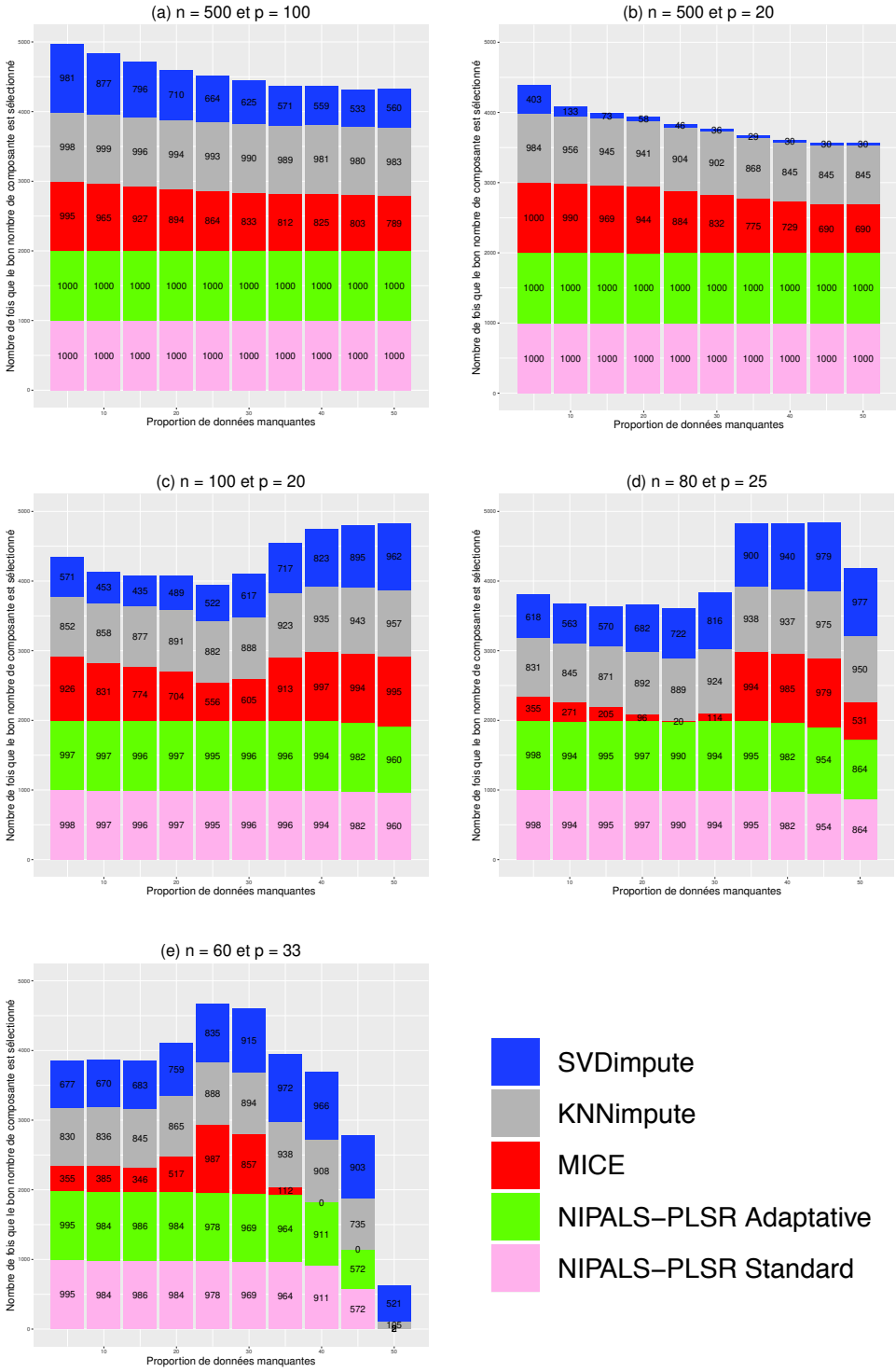


Figure B.1 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

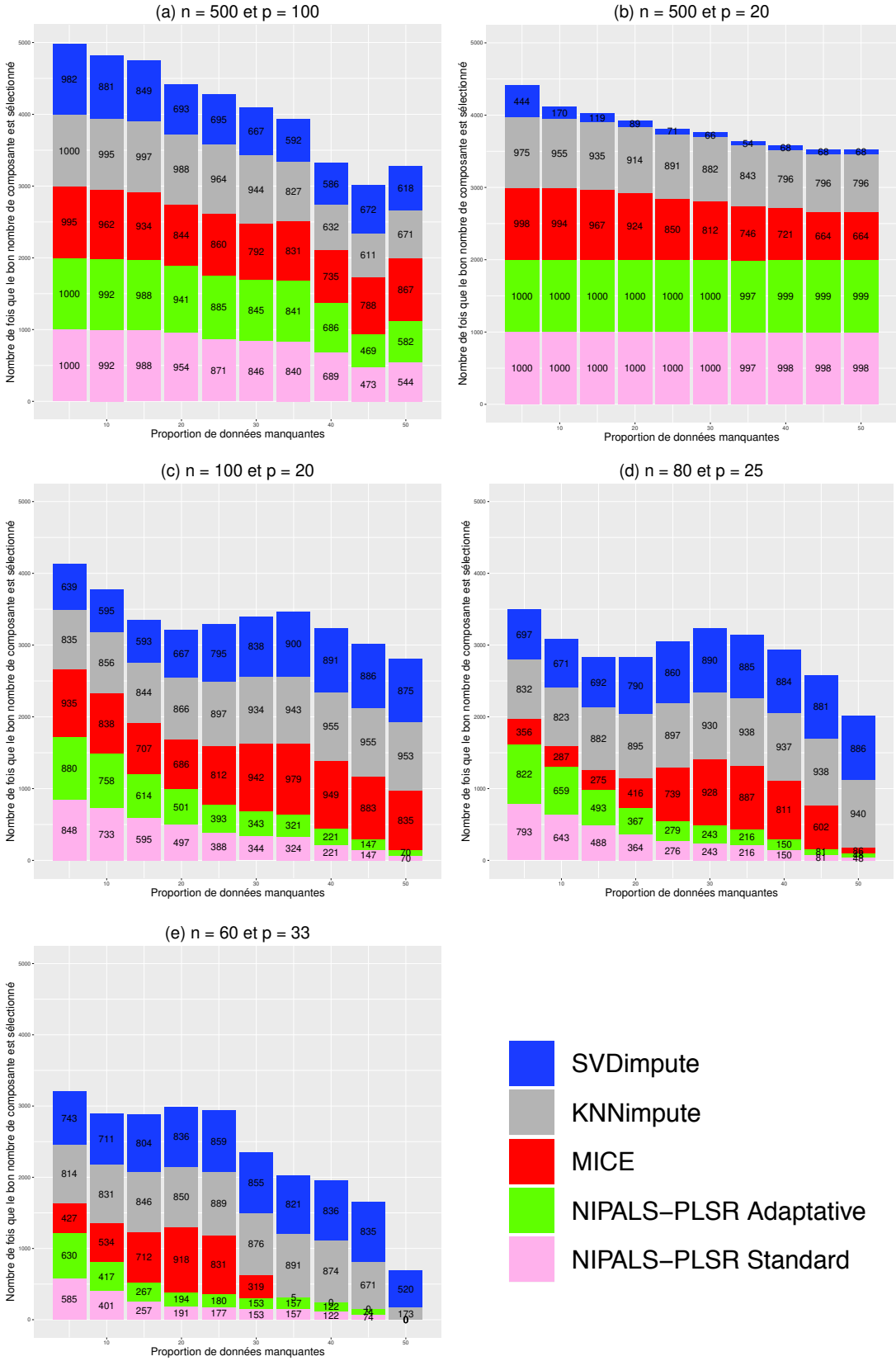


Figure B.2 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

Q^2-10-*Fold*

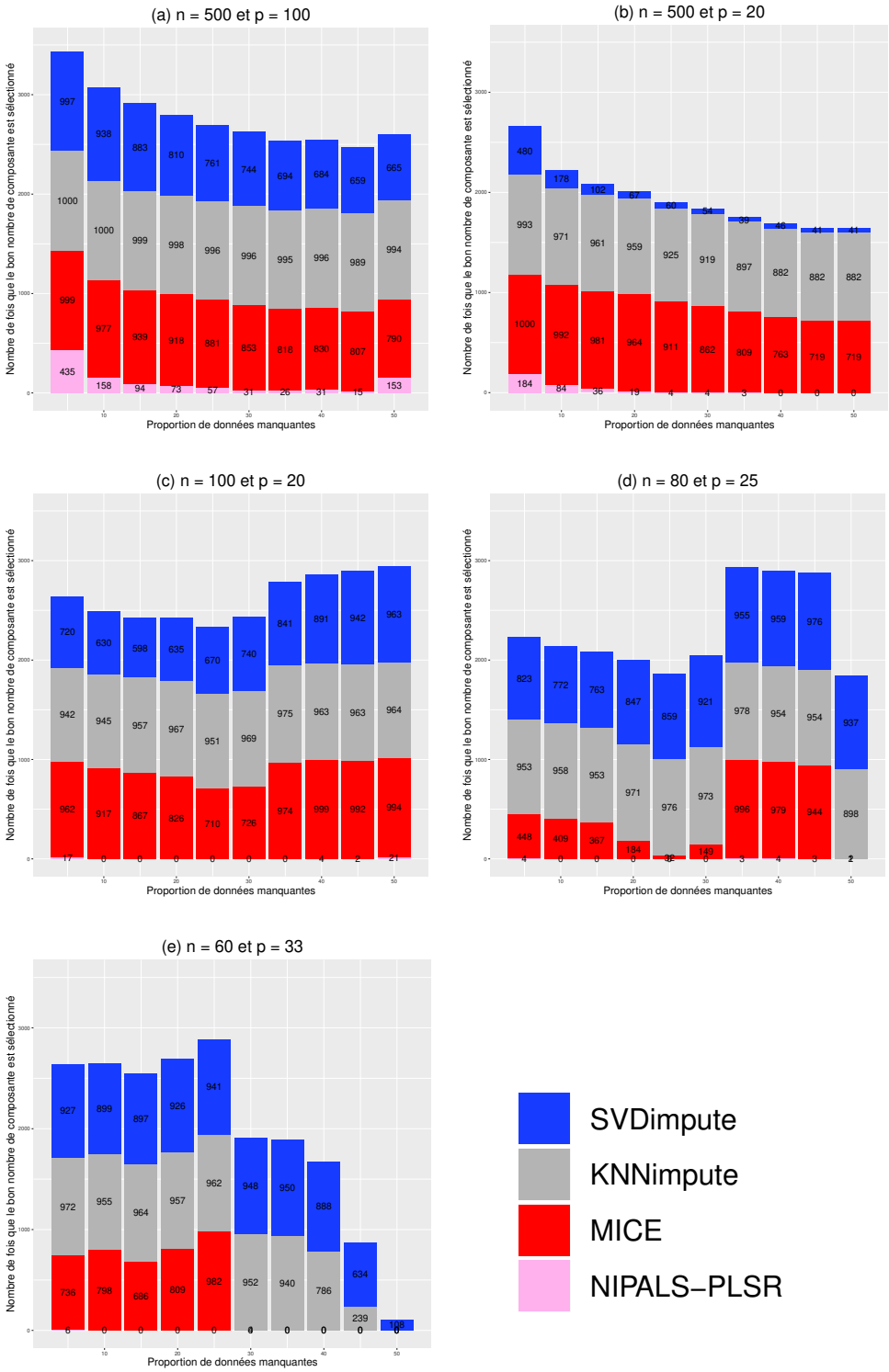


Figure B.3 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

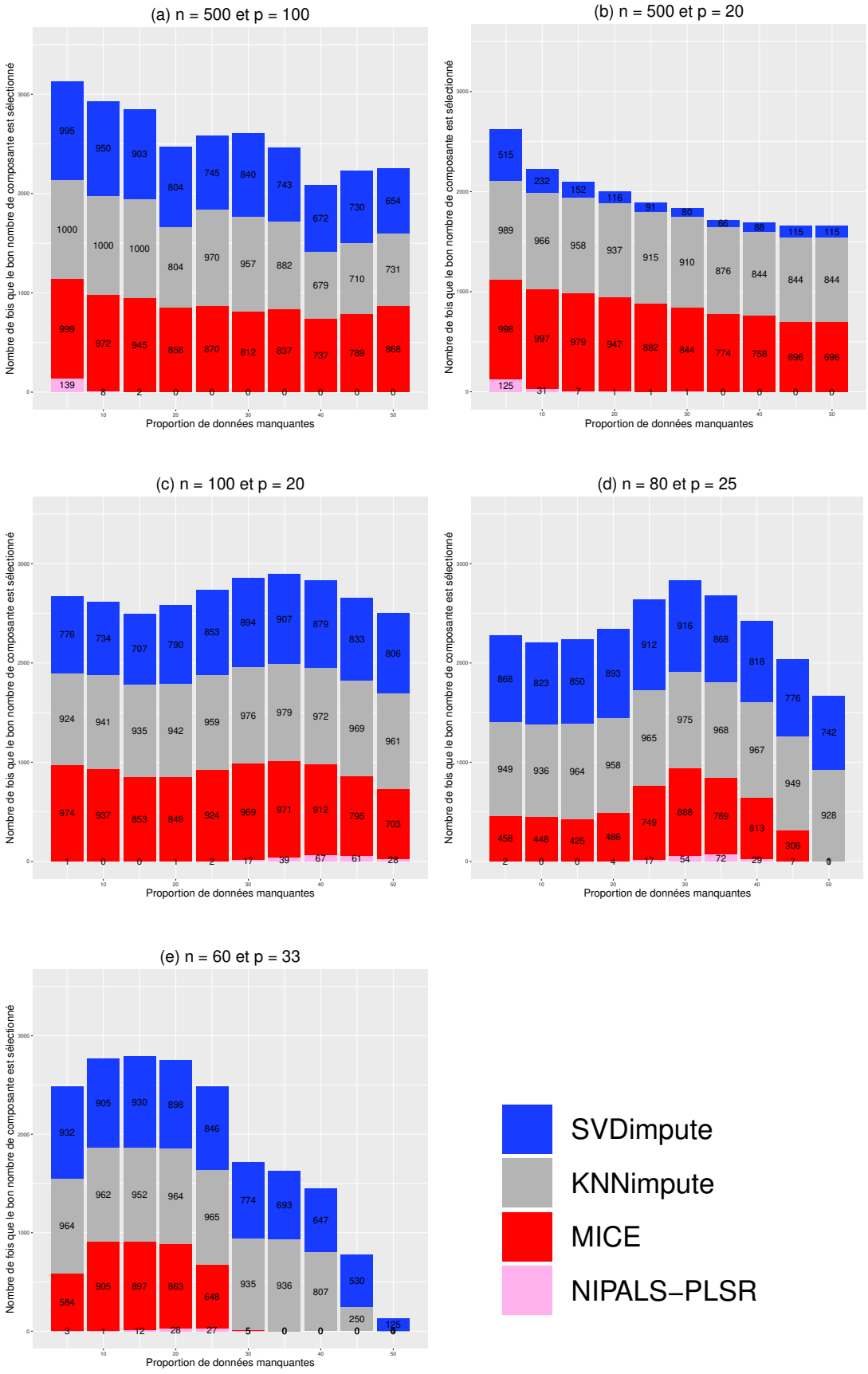


Figure B.4 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC

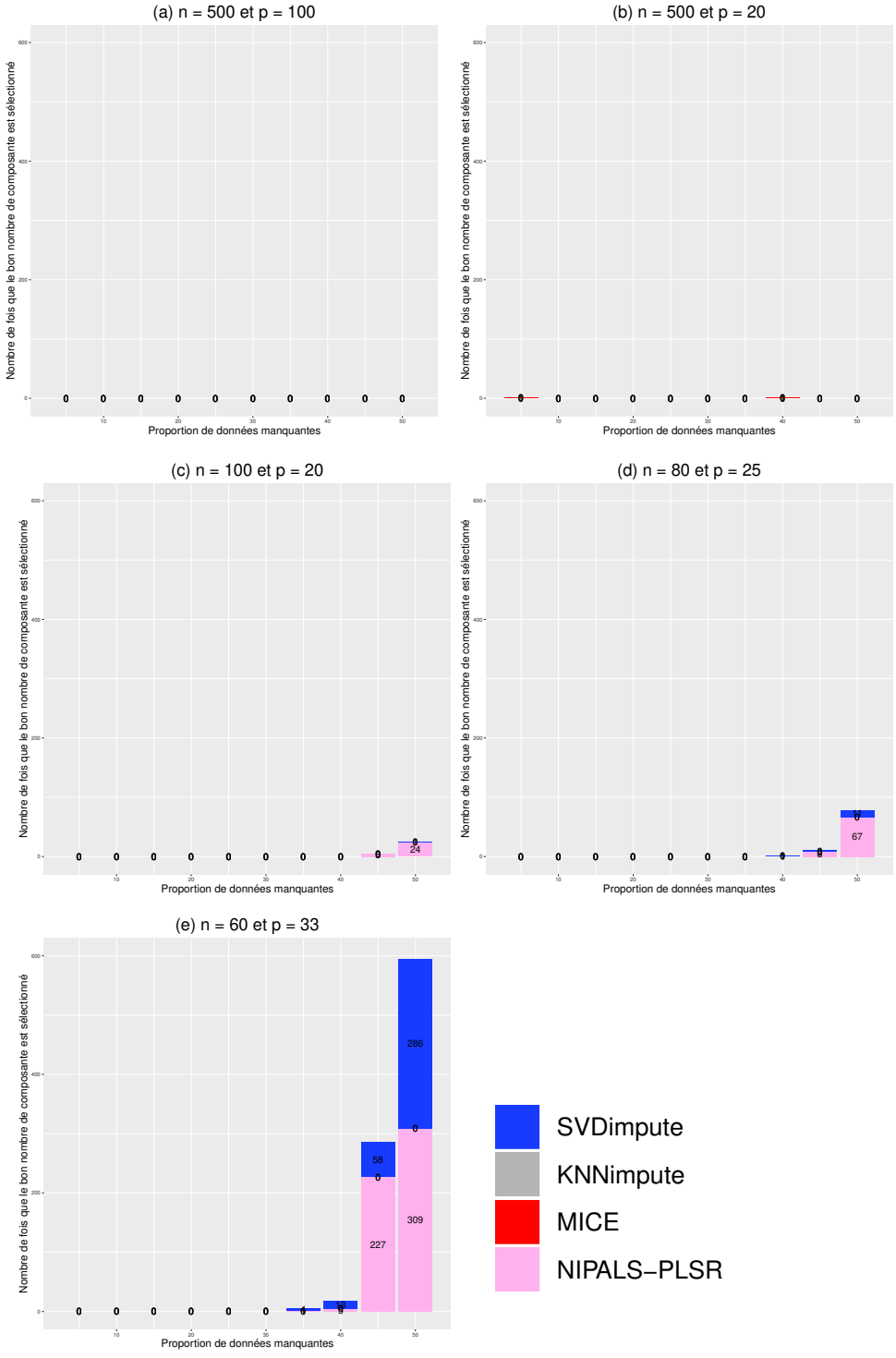


Figure B.5 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

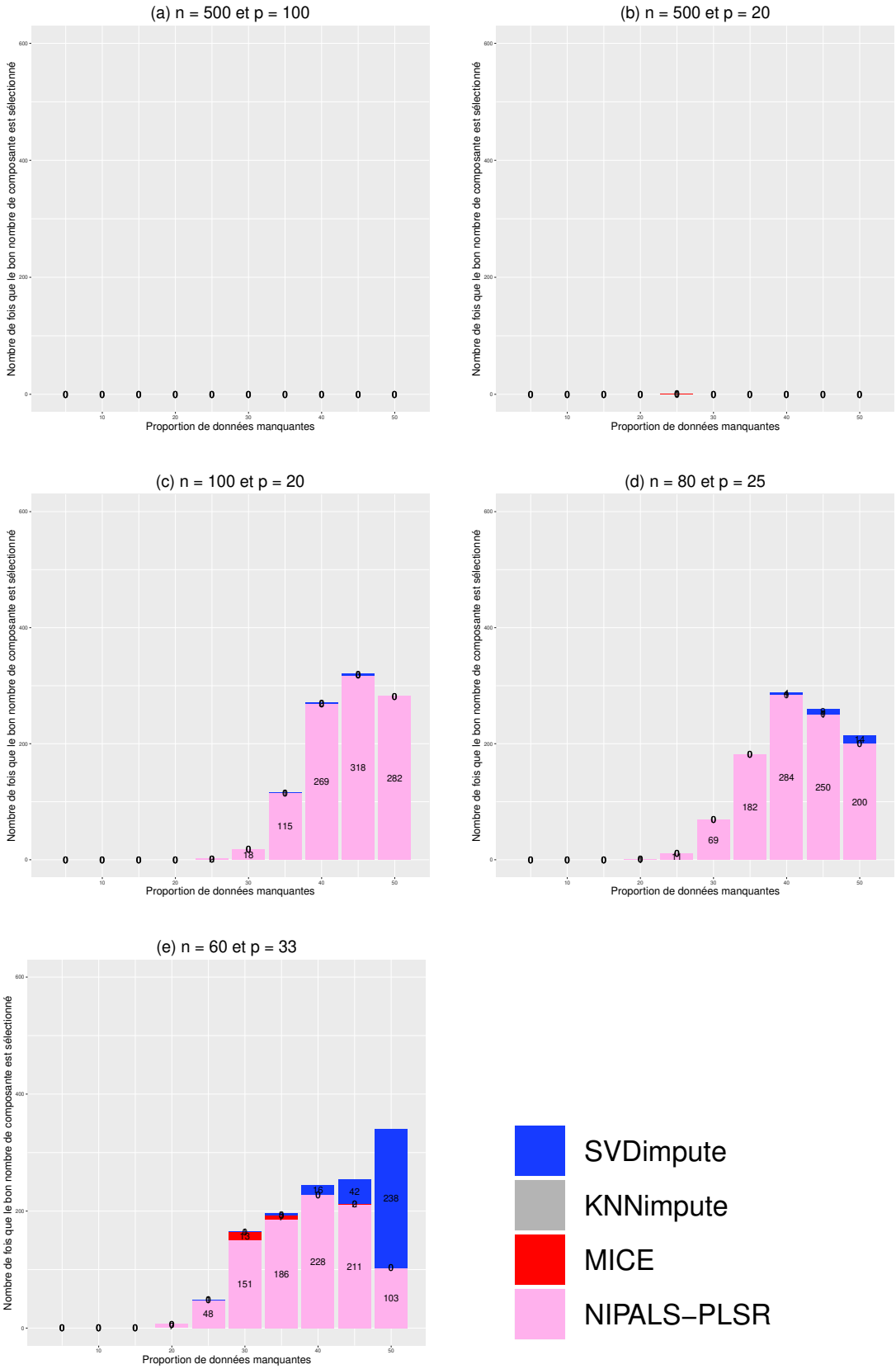


Figure B.6 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC-DoF

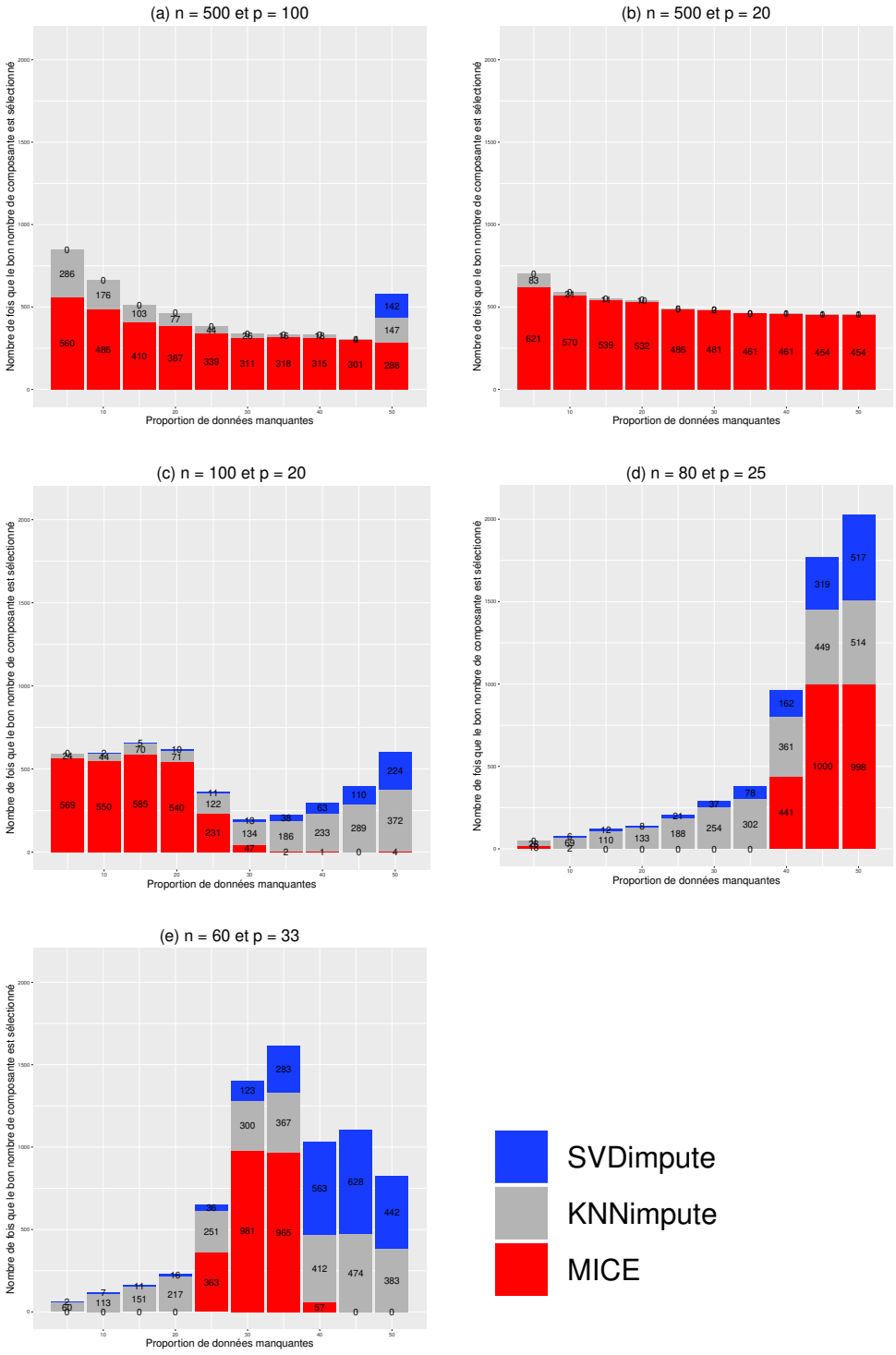


Figure B.7 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

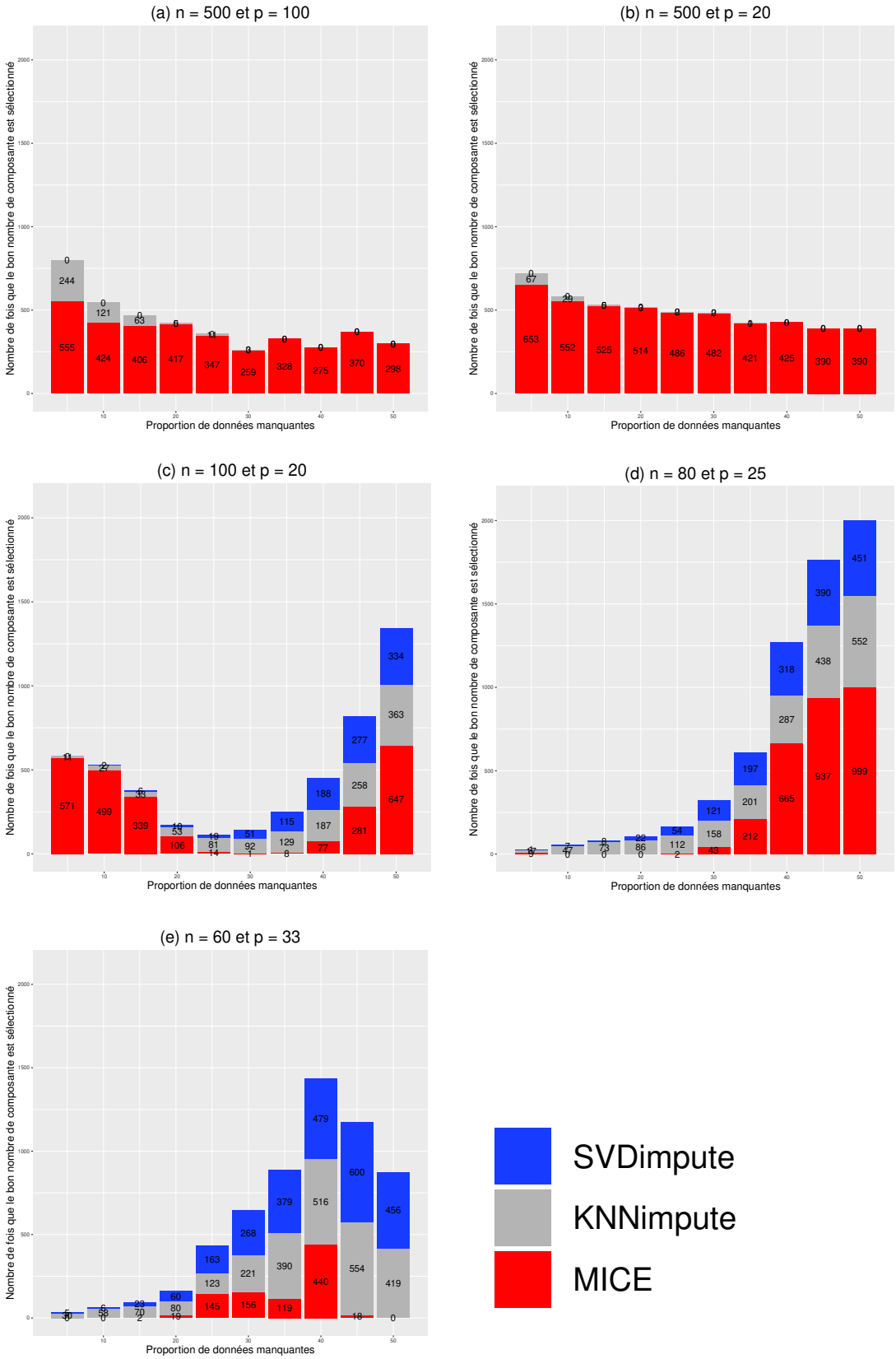


Figure B.8 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC

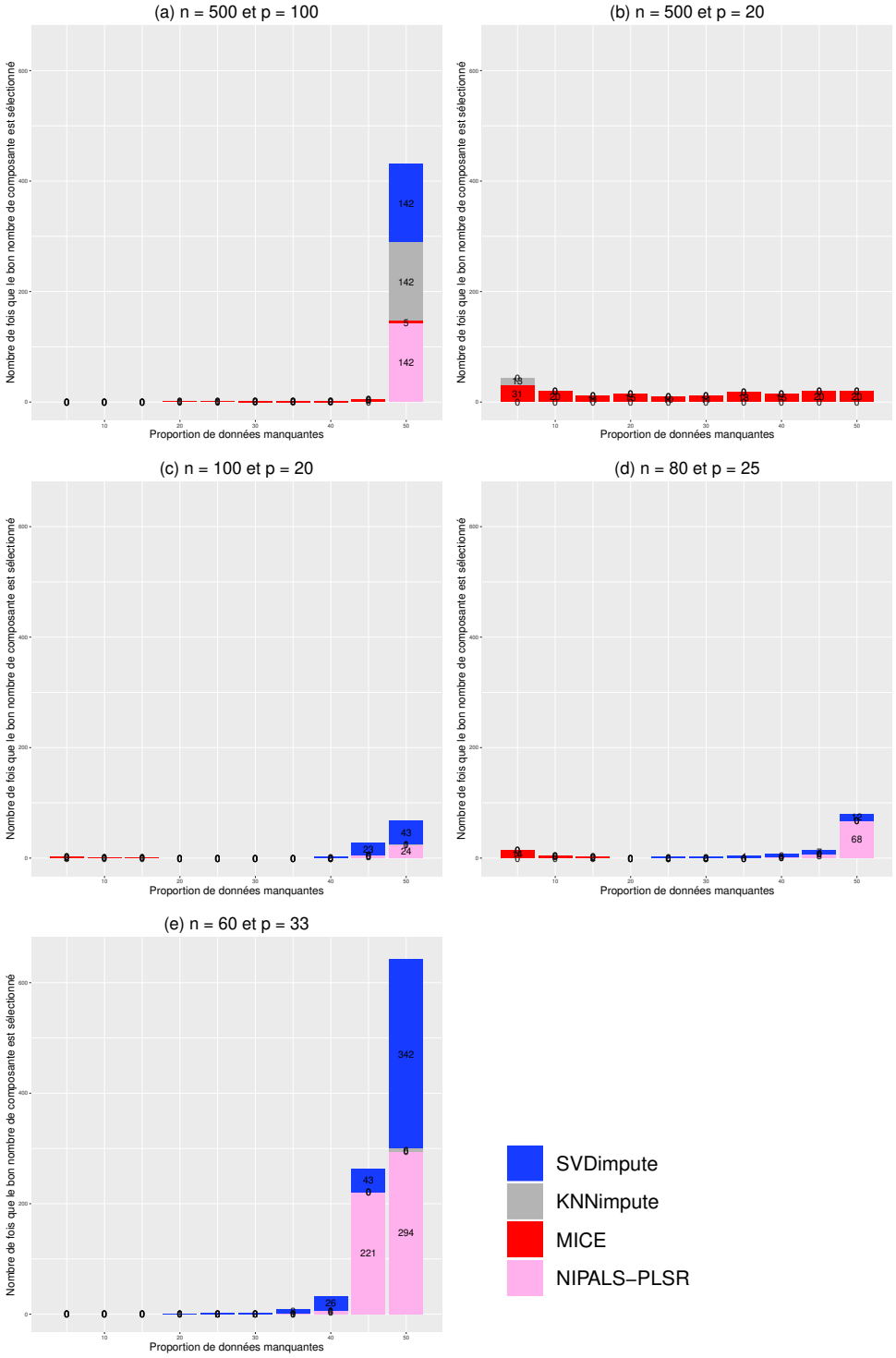


Figure B.9 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

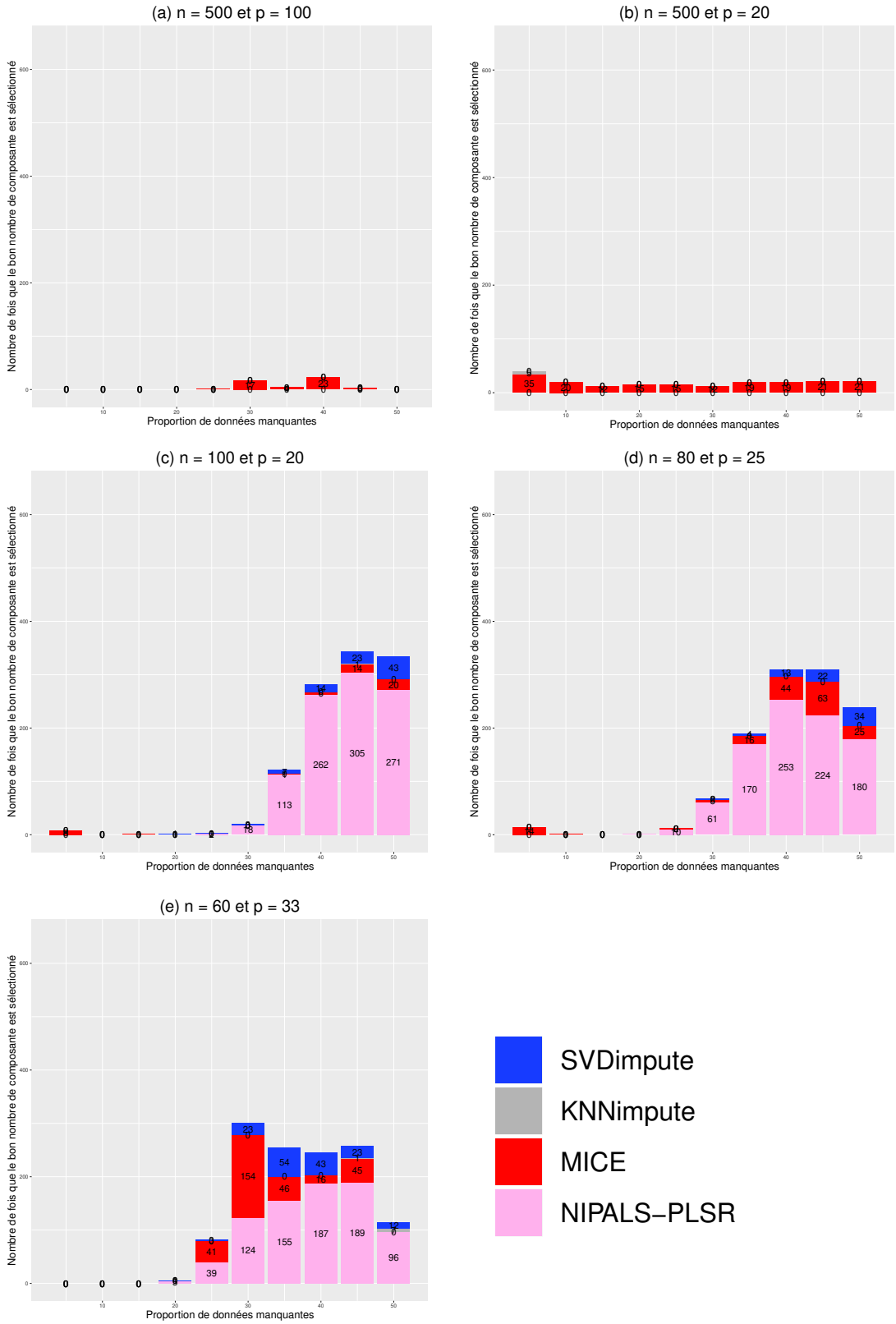


Figure B.10 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC-DoF

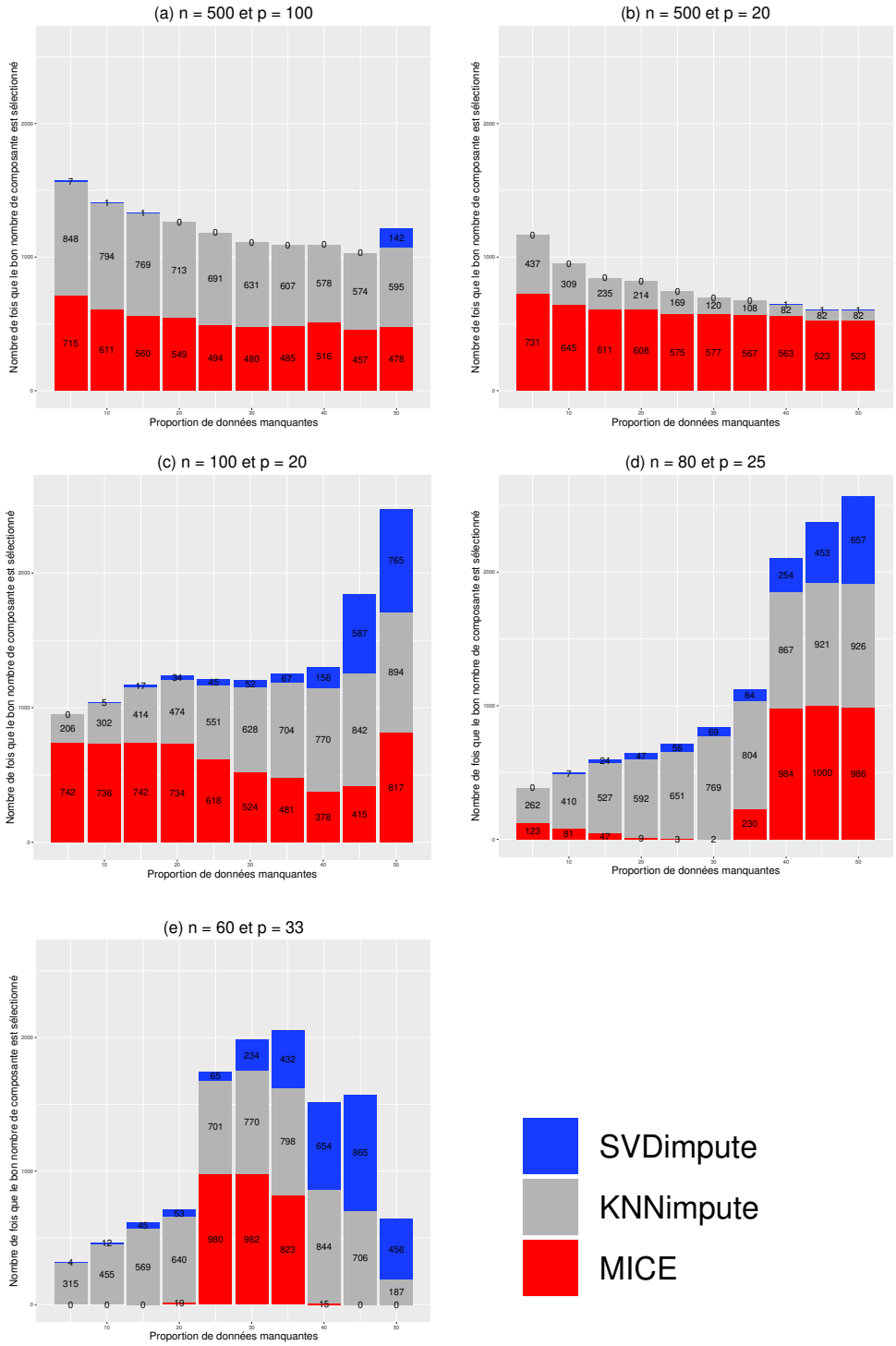


Figure B.11 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

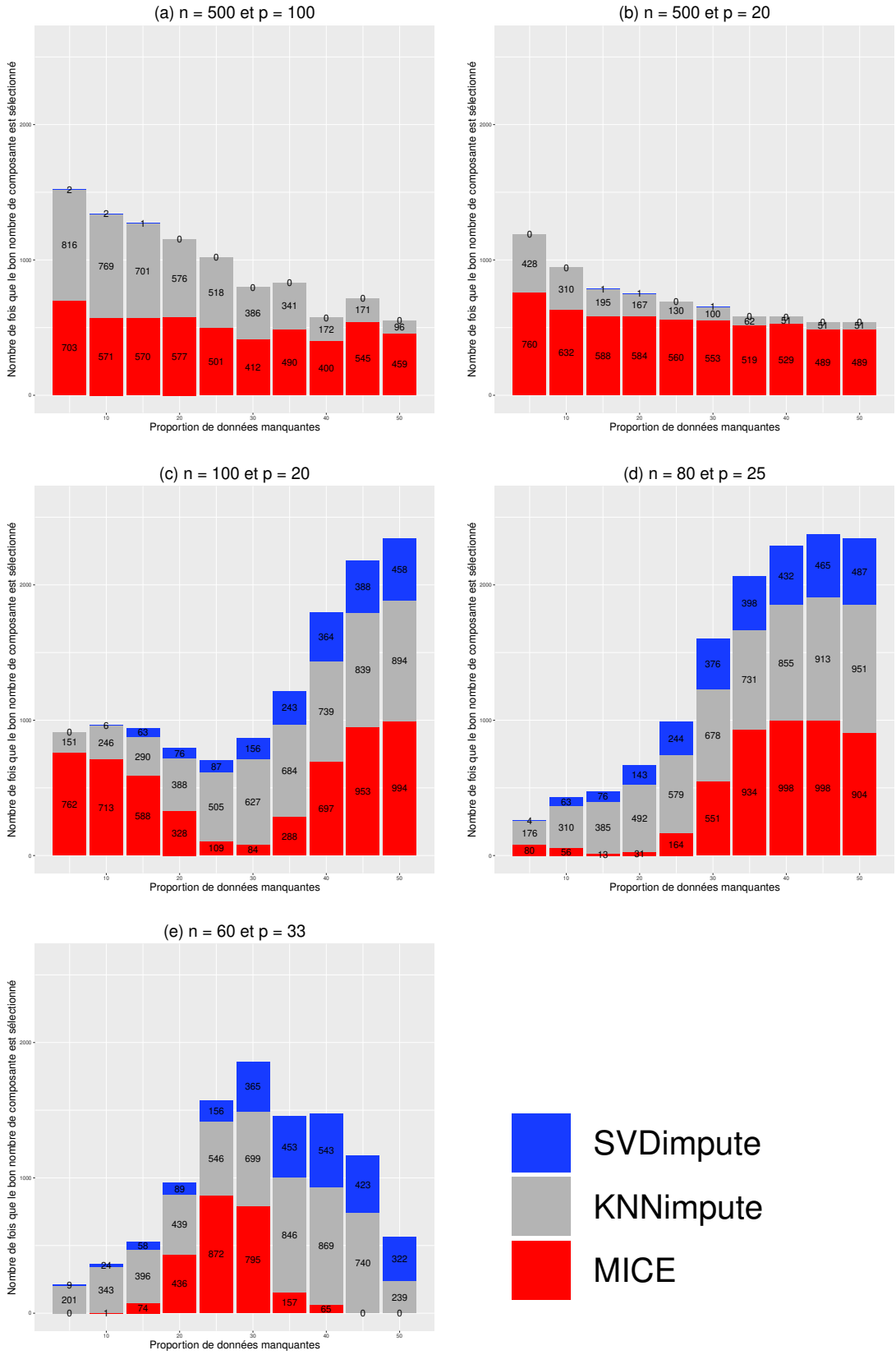


Figure B.12 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

B.1.2 Nombre vrai de composantes = 4

Q^2-LOO

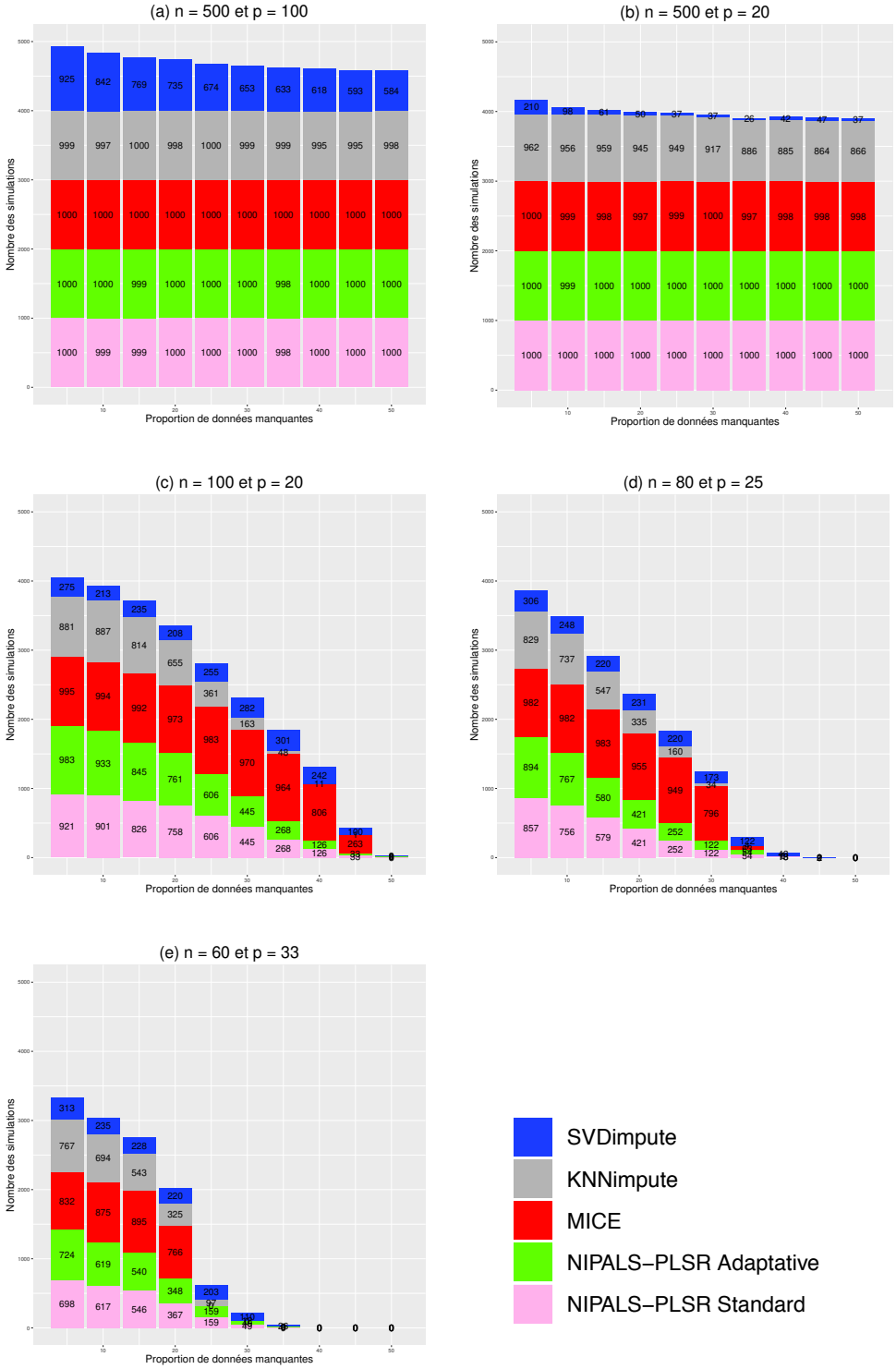


Figure B.13 – Nombre de choix corrects du nombre de composantes avec le critère *Q^2-LOO* selon la dimension des données, les méthodes et t^* = 4 sur 1000 simulations, sous l'hypothèse *MCAR*.

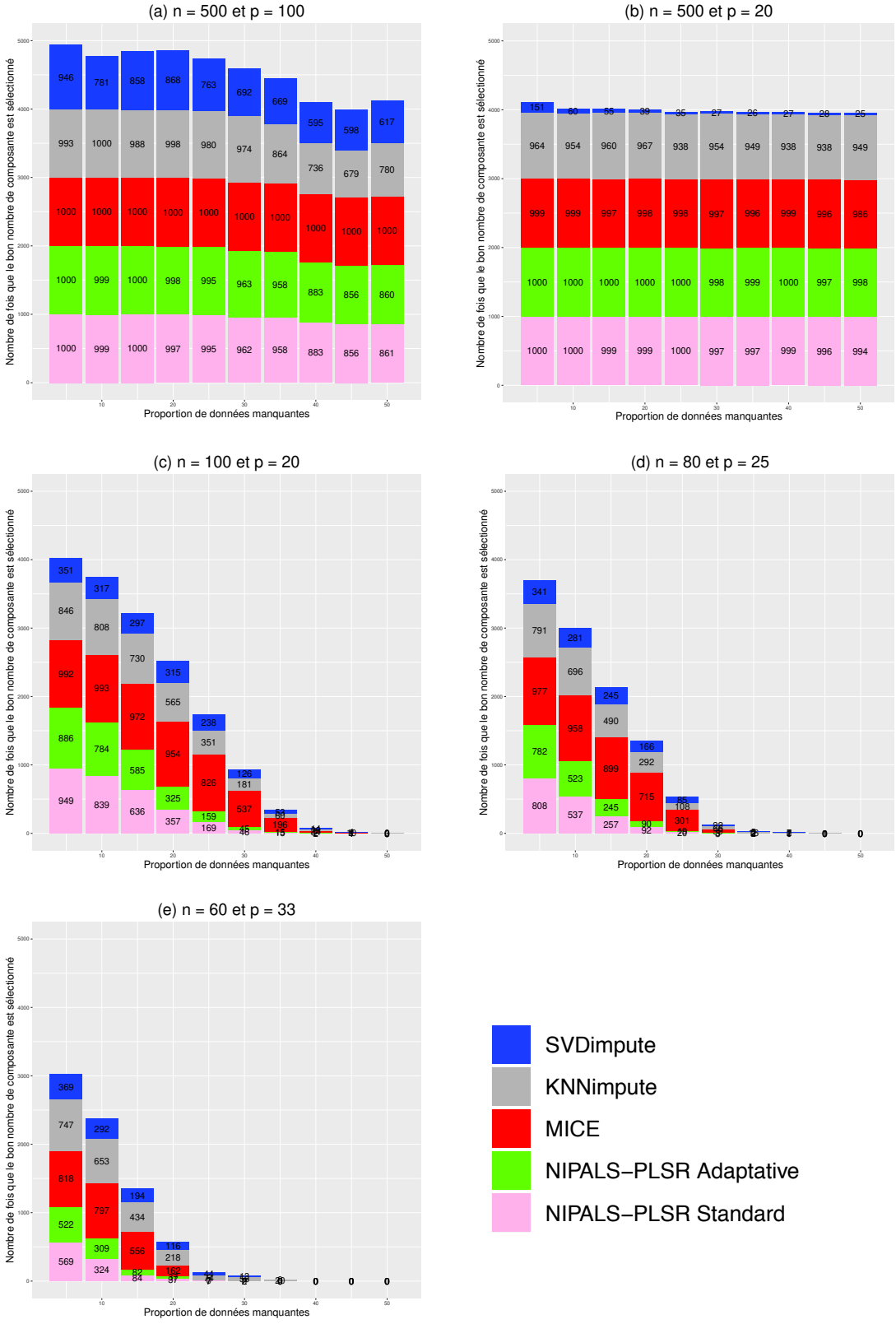


Figure B.14 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

Q^2-10-*Fold*

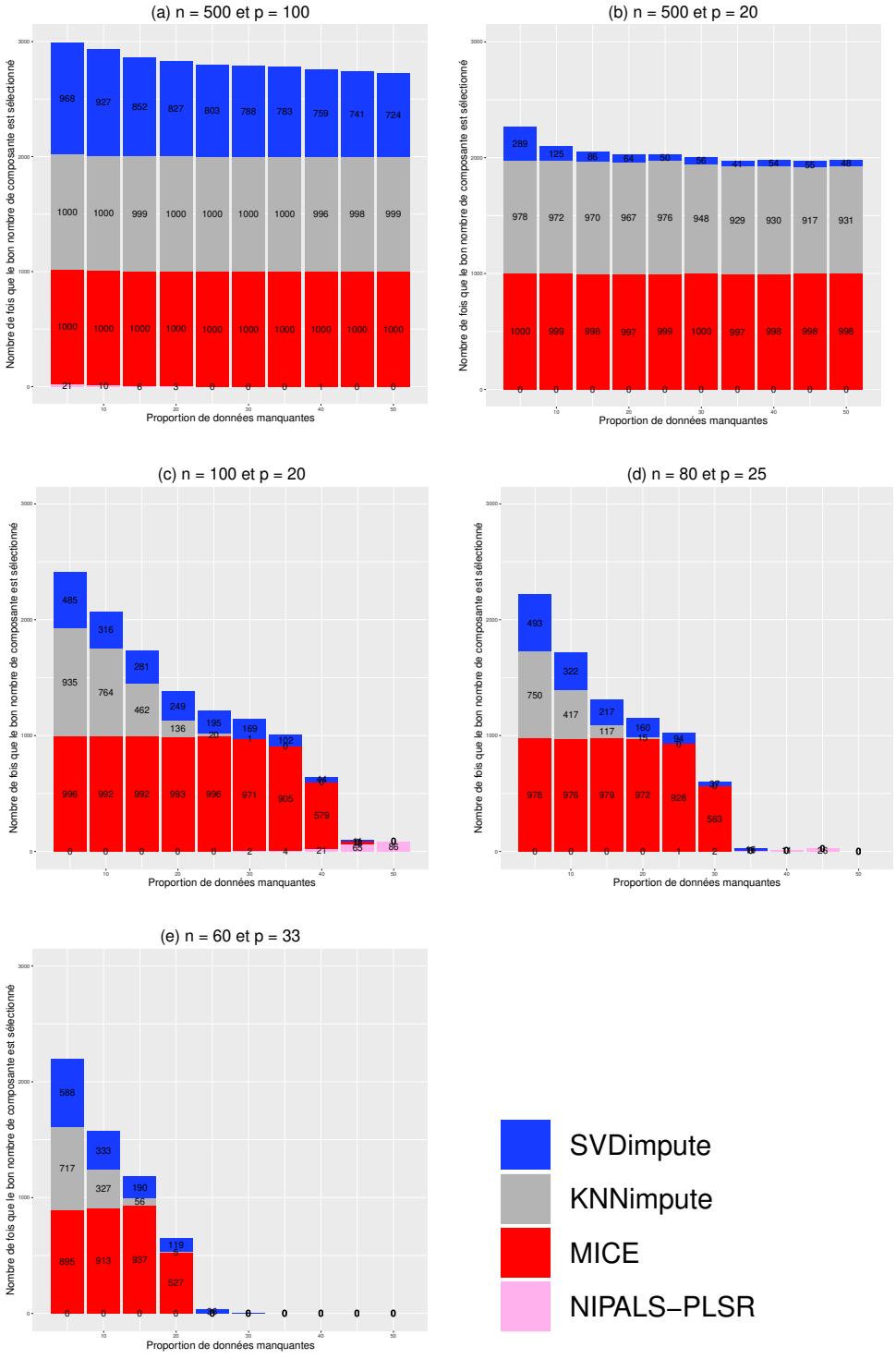


Figure B.15 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

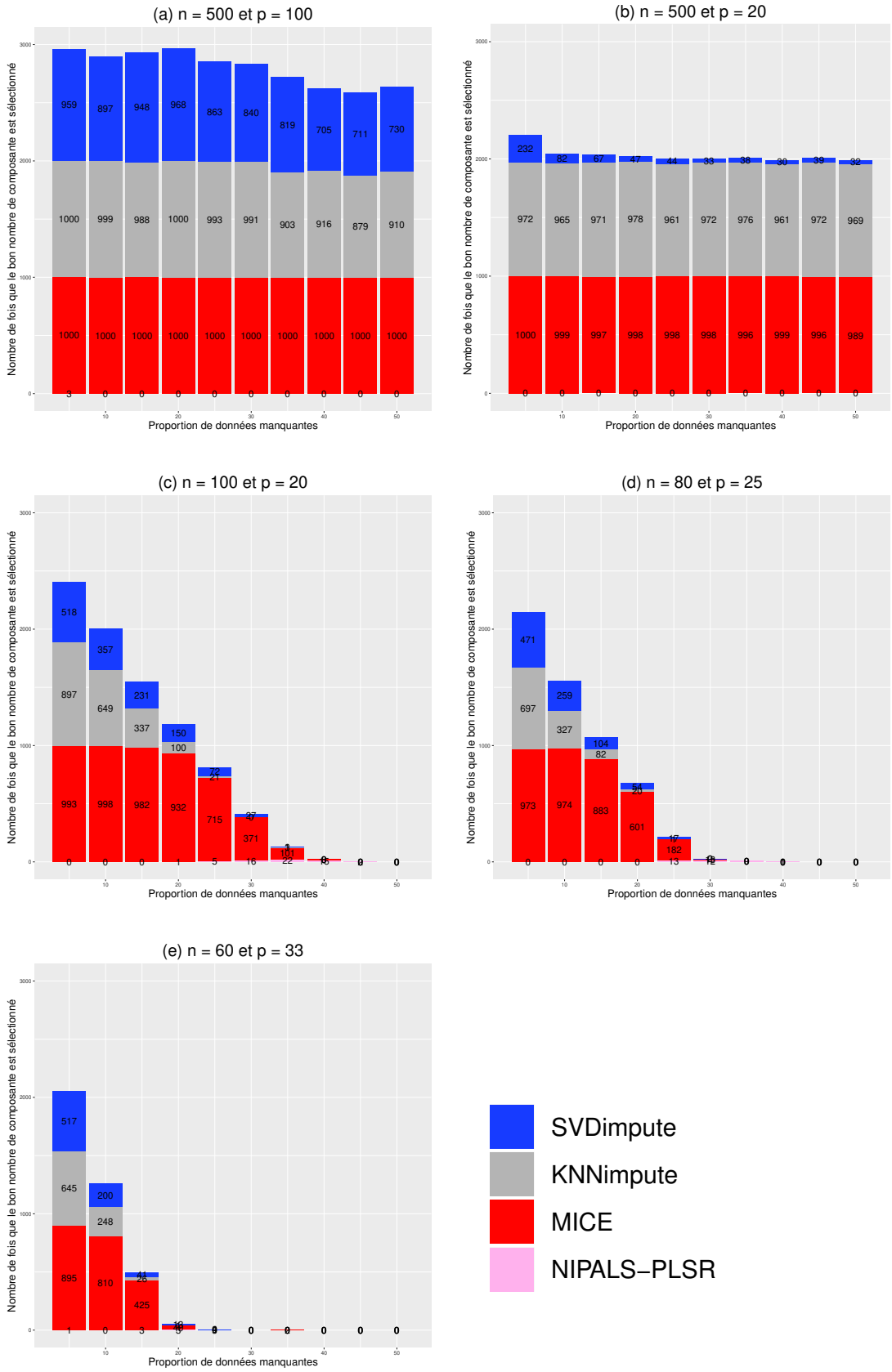


Figure B.16 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC

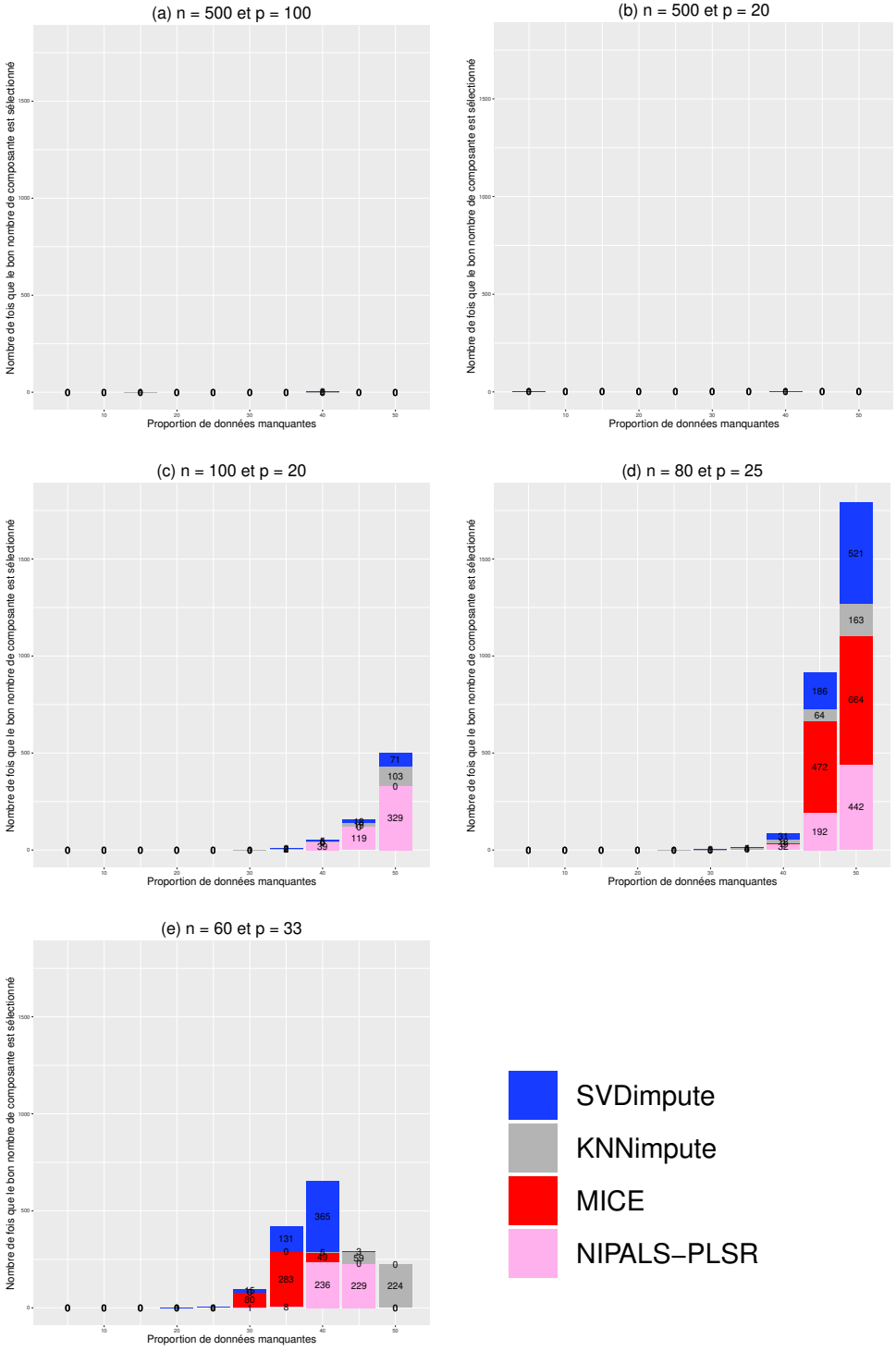


Figure B.17 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

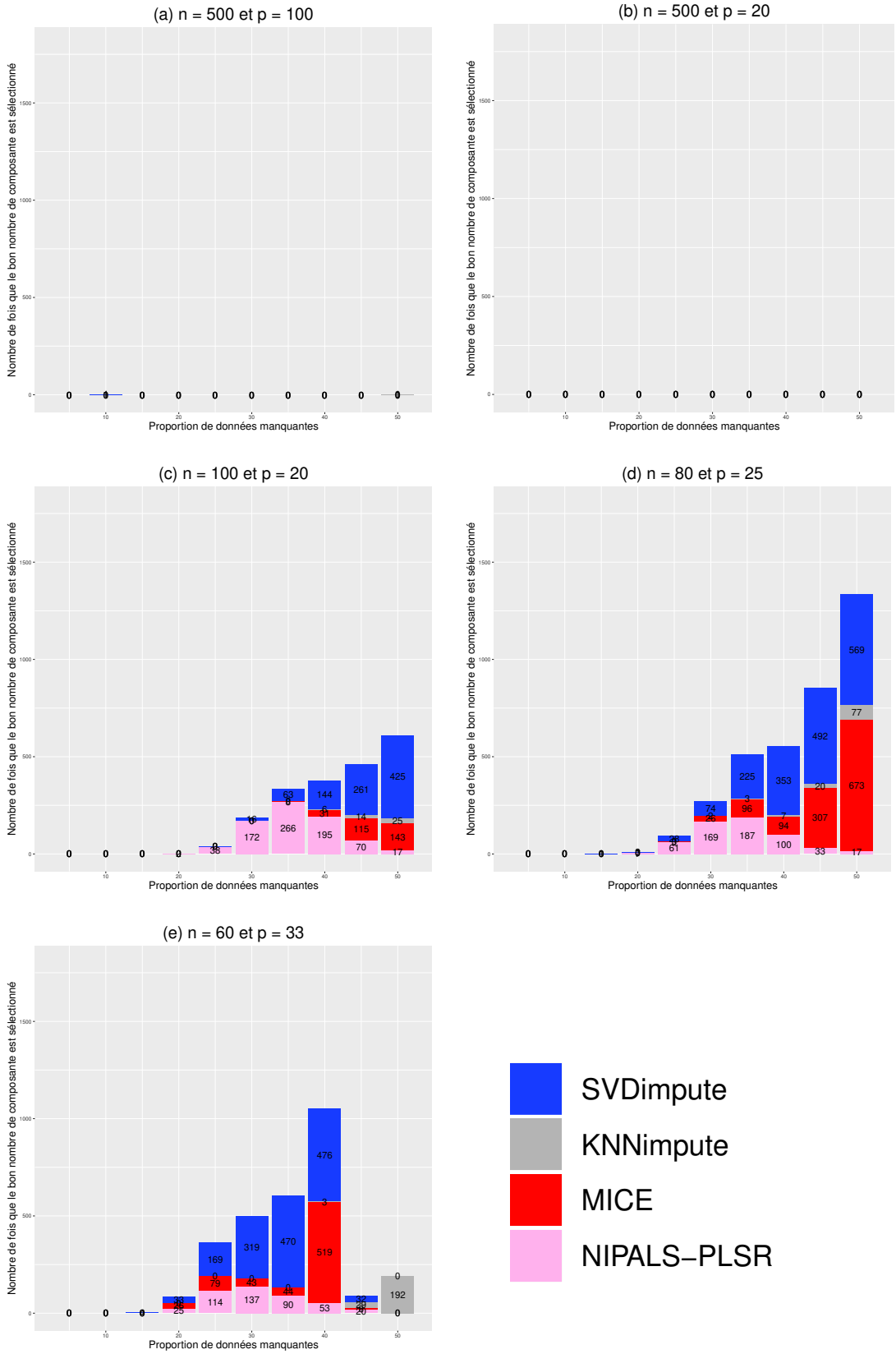


Figure B.18 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC-DoF

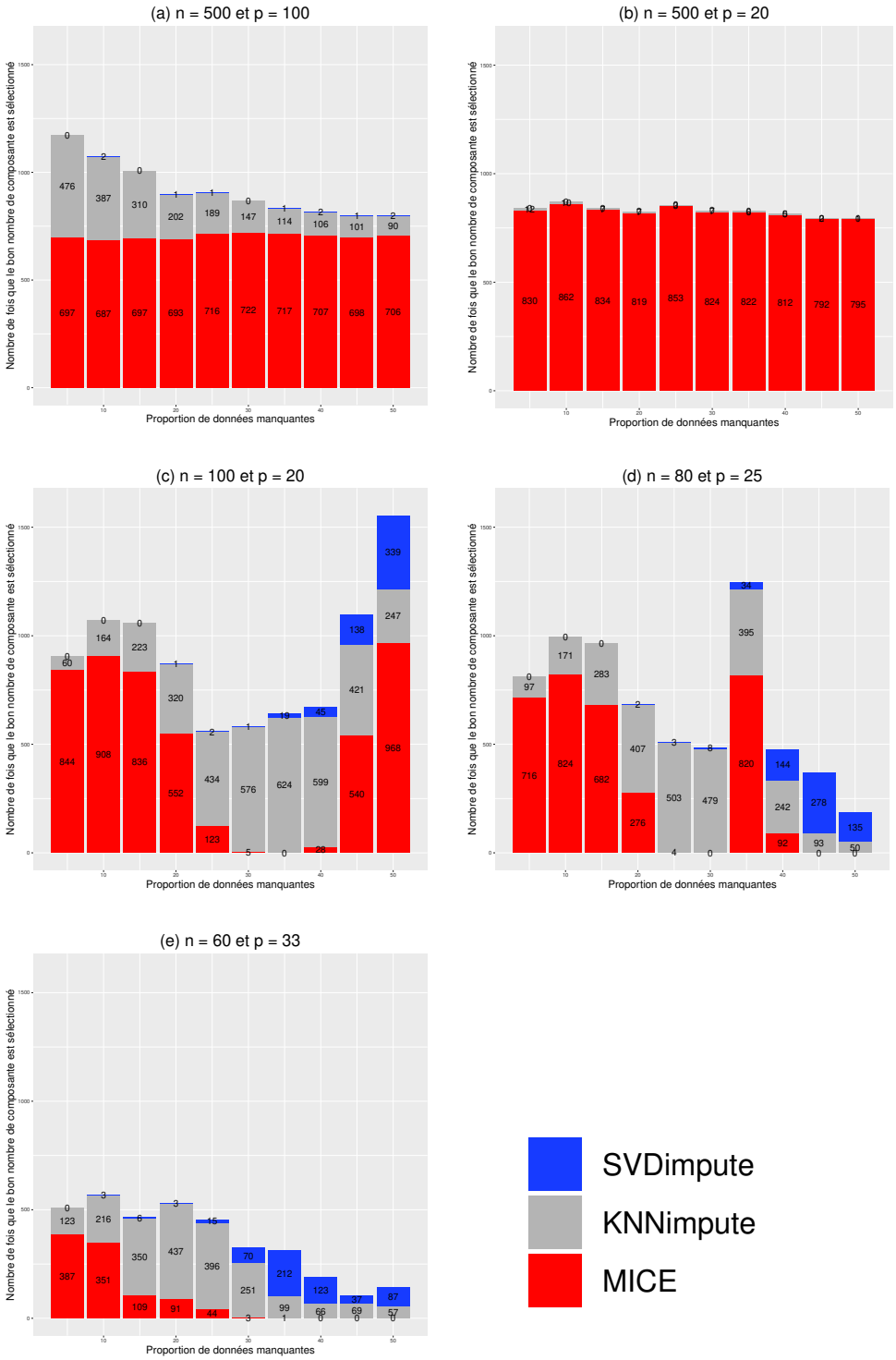


Figure B.19 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

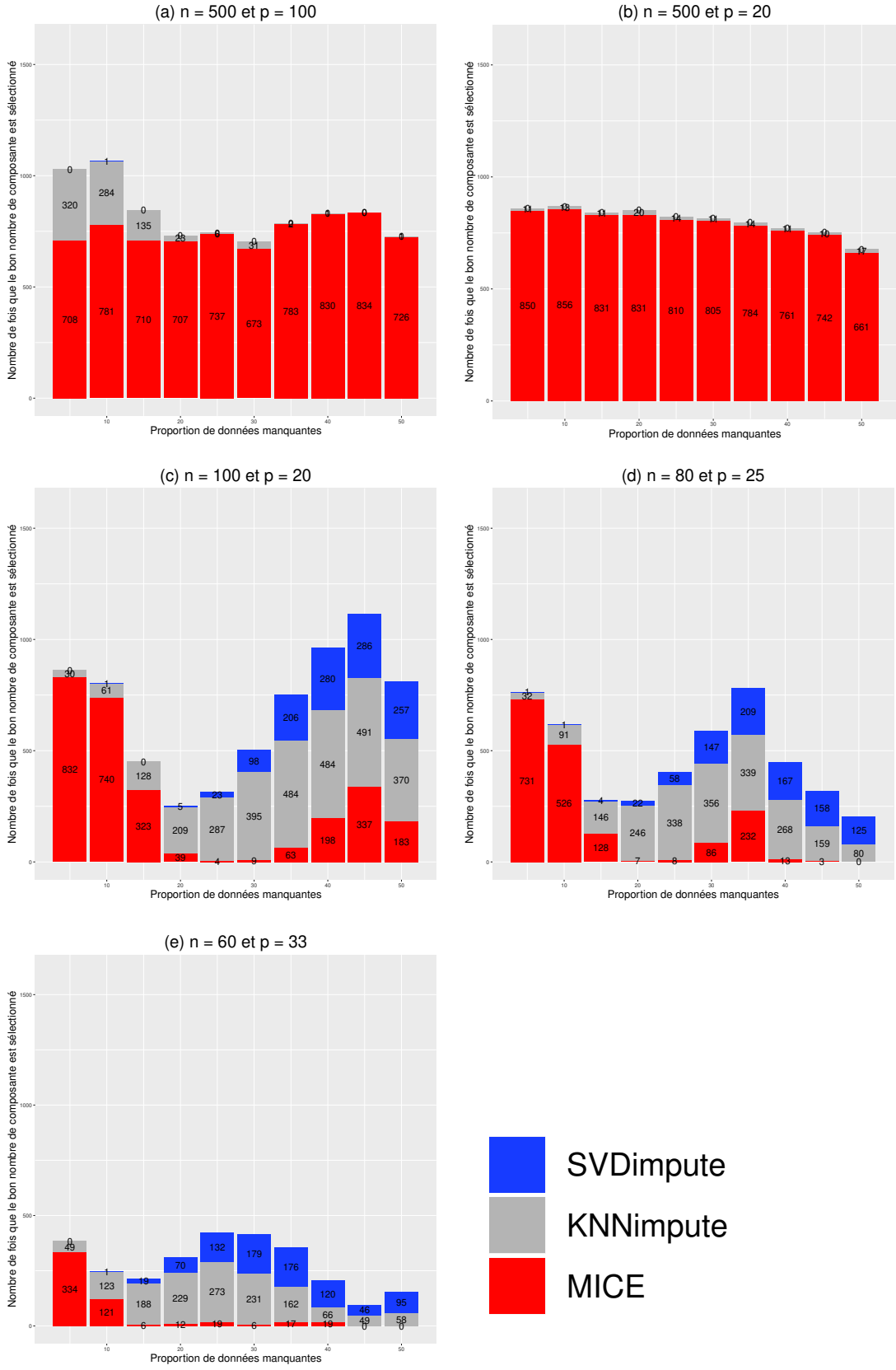


Figure B.20 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC

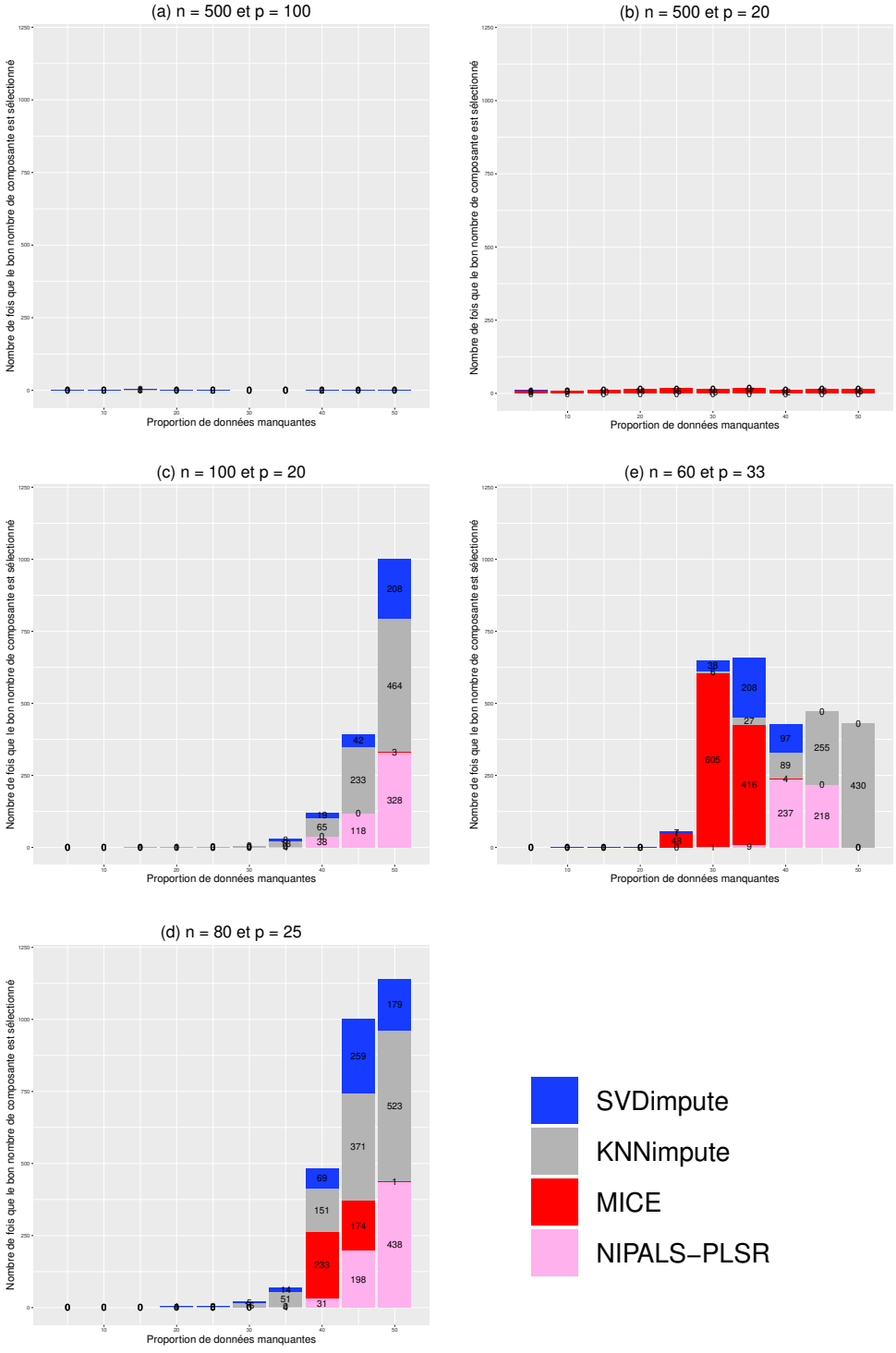


Figure B.21 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

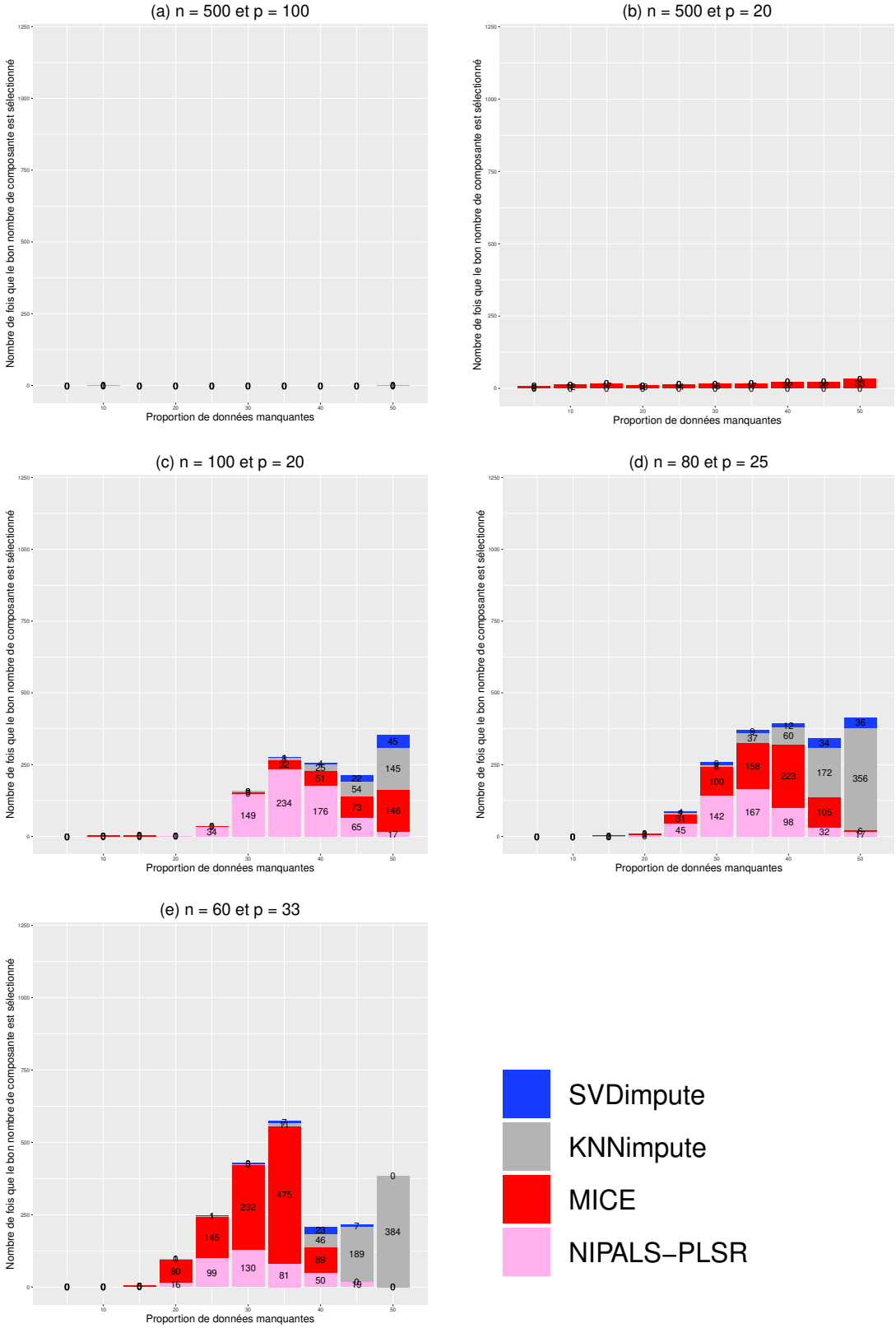


Figure B.22 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC-DoF

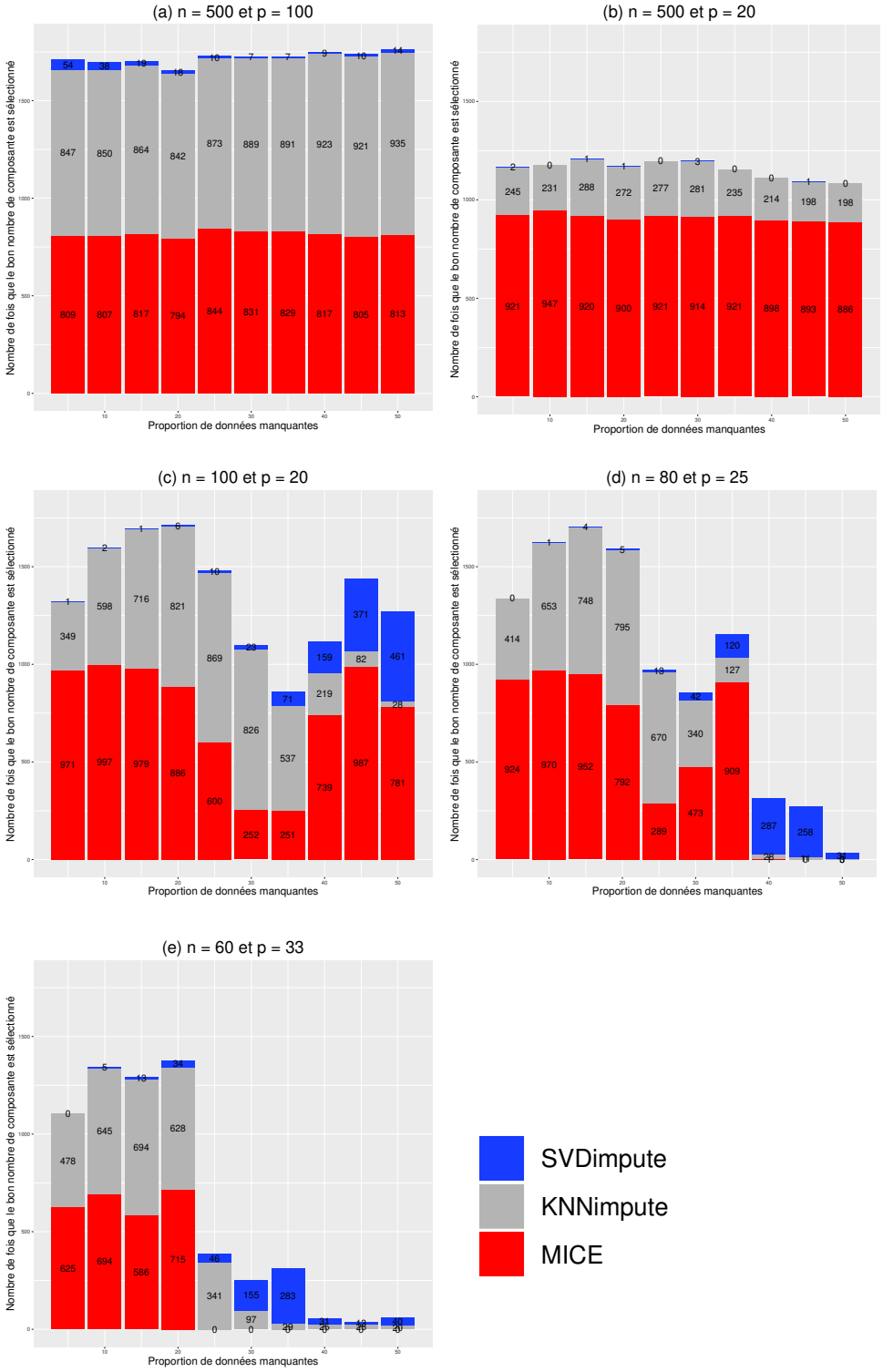


Figure B.23 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

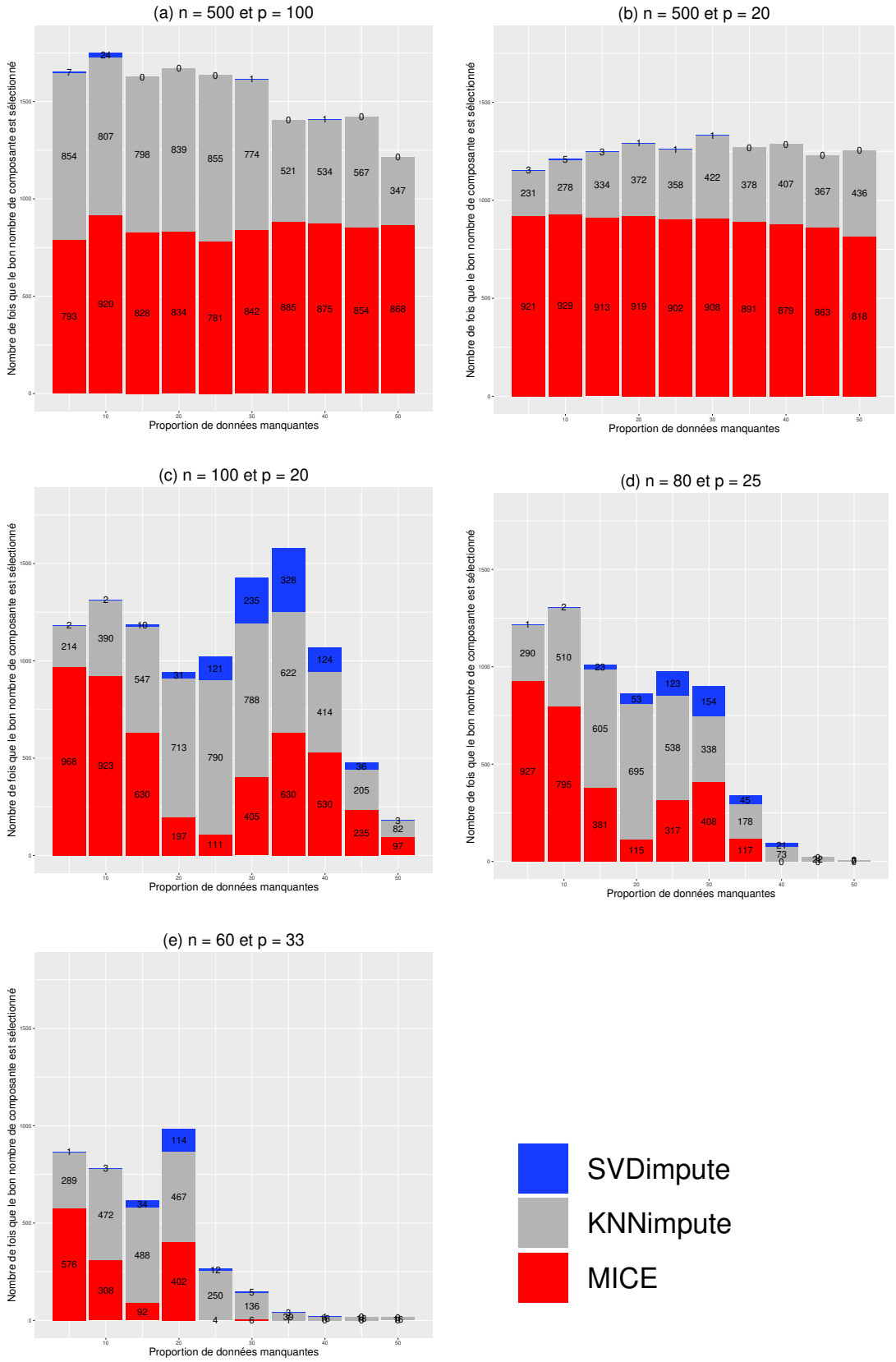


Figure B.24 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

B.1.3 Nombre vrai de composantes = 6

Q^2-LOO

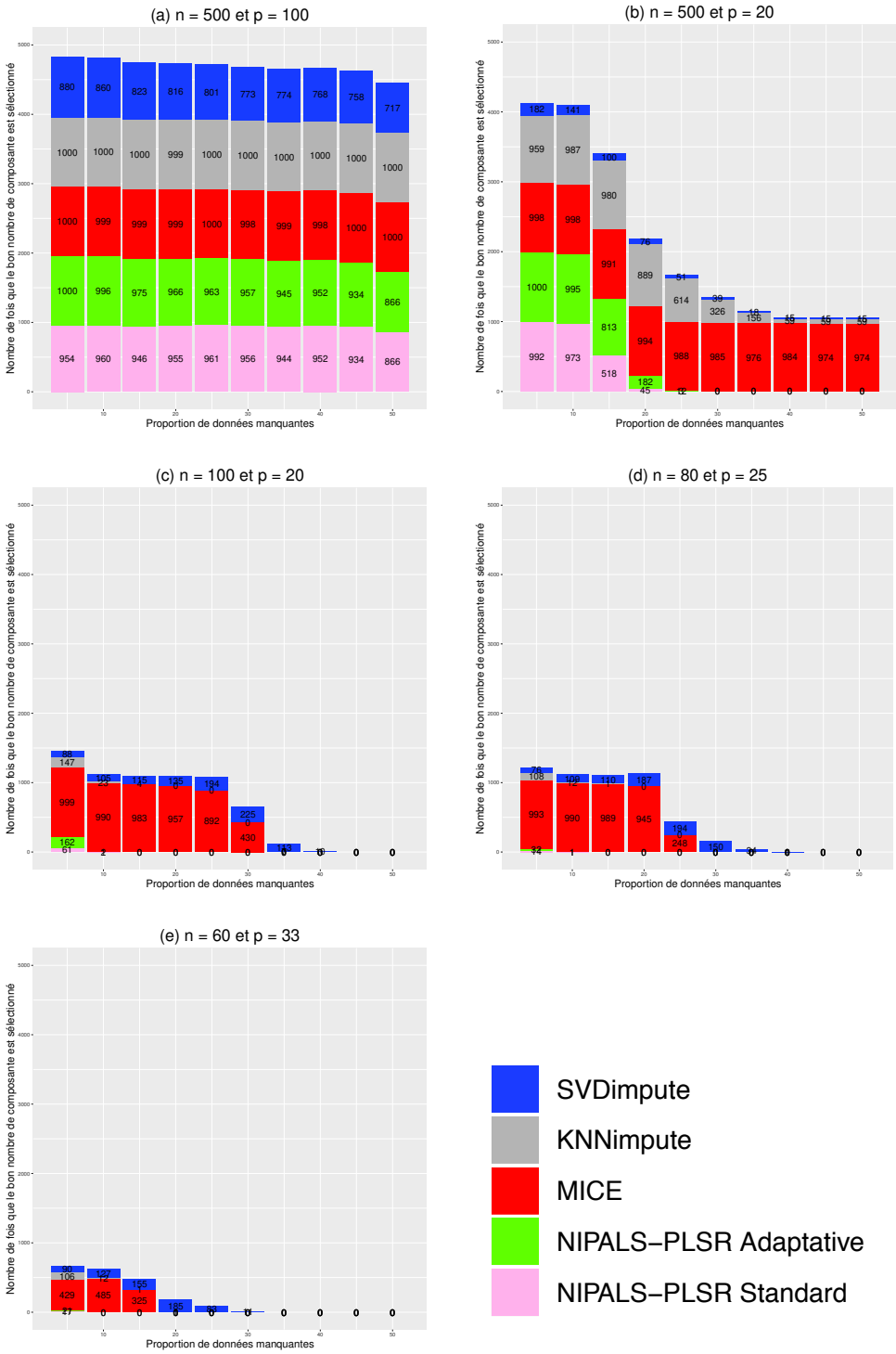


Figure B.25 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

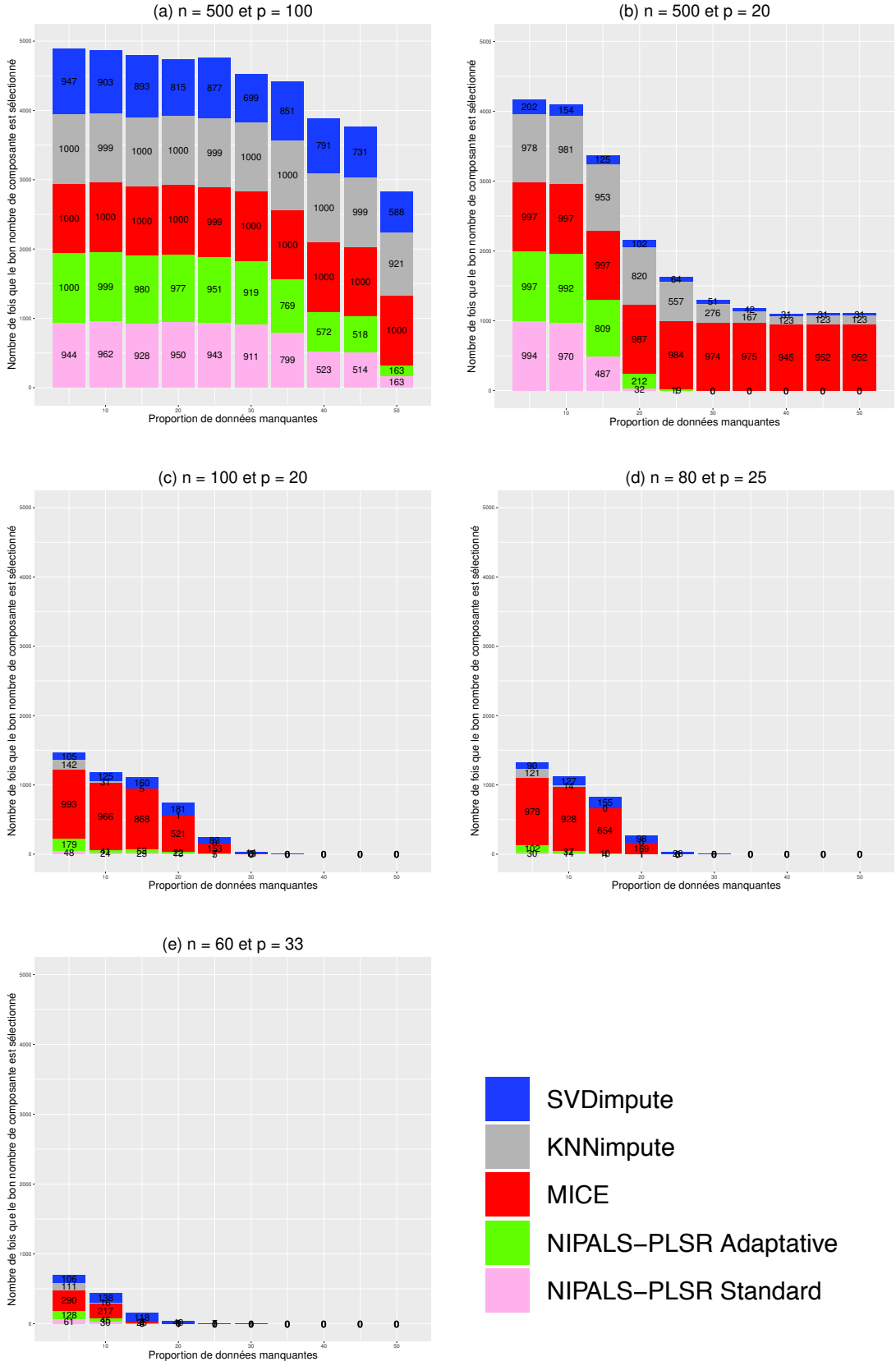


Figure B.26 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

Q^2-10-*Fold*

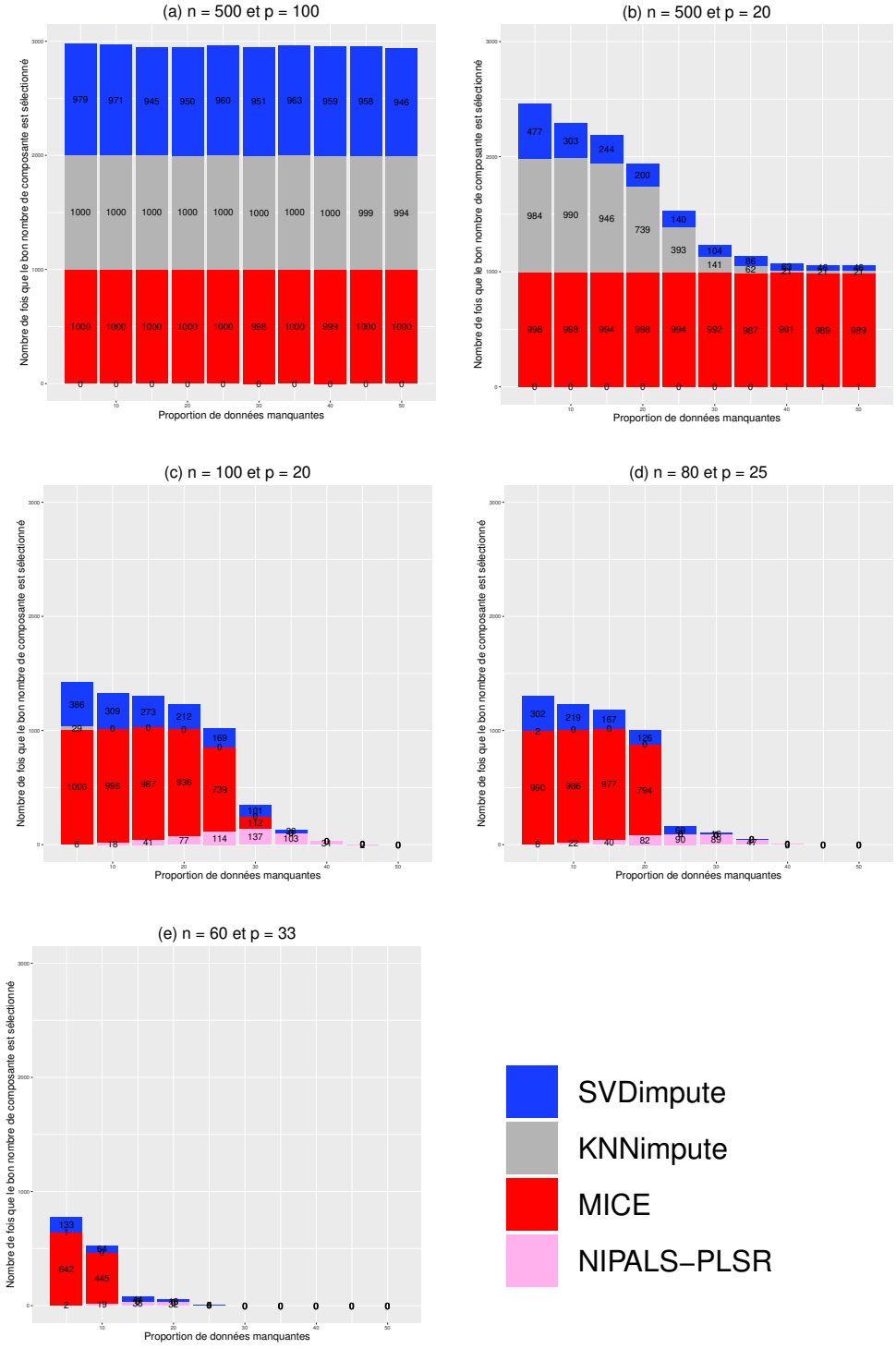


Figure B.27 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et t^* = 6 sur 1000 simulations, sous l'hypothèse *MCAR*.

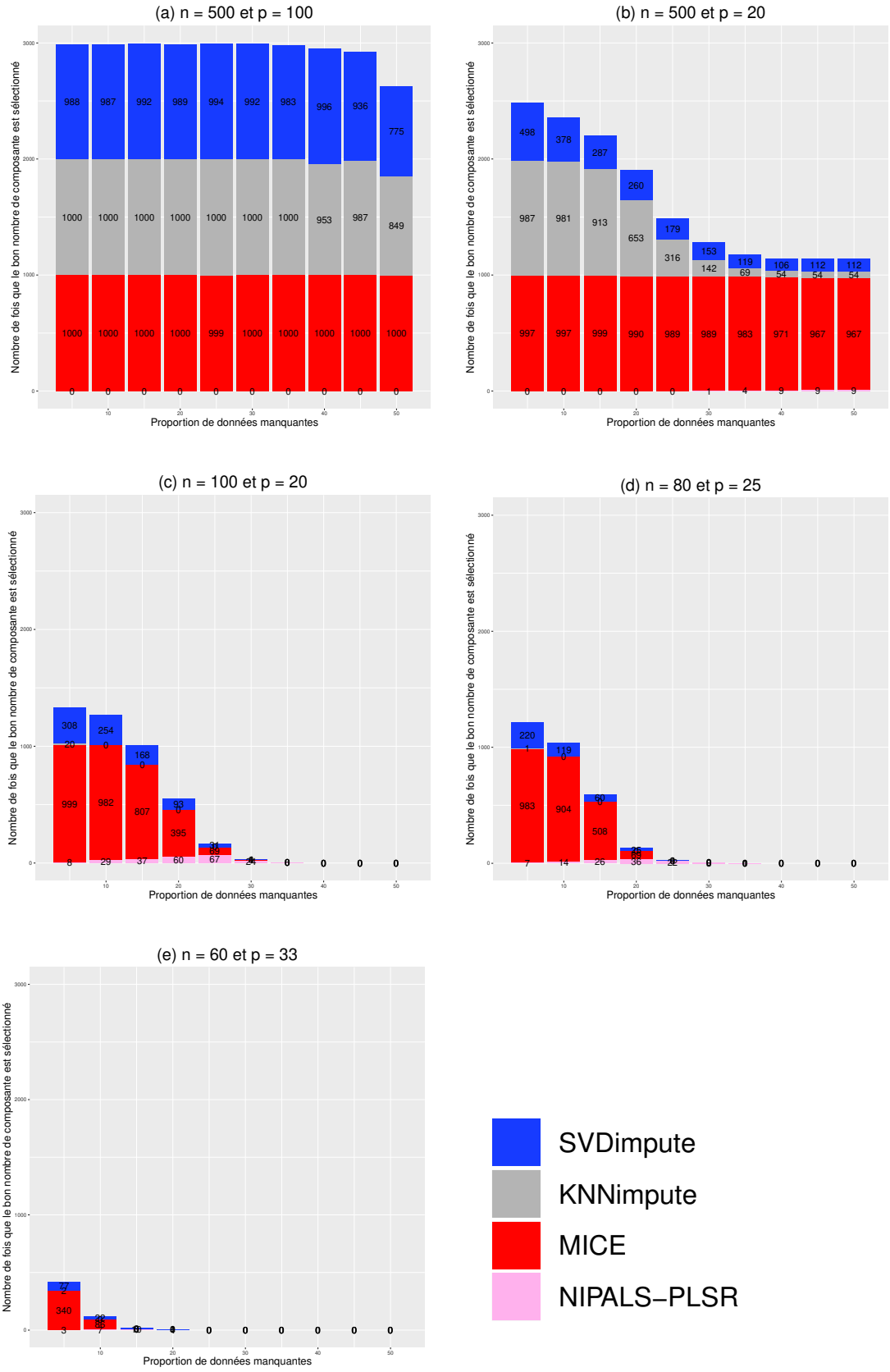


Figure B.28 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC

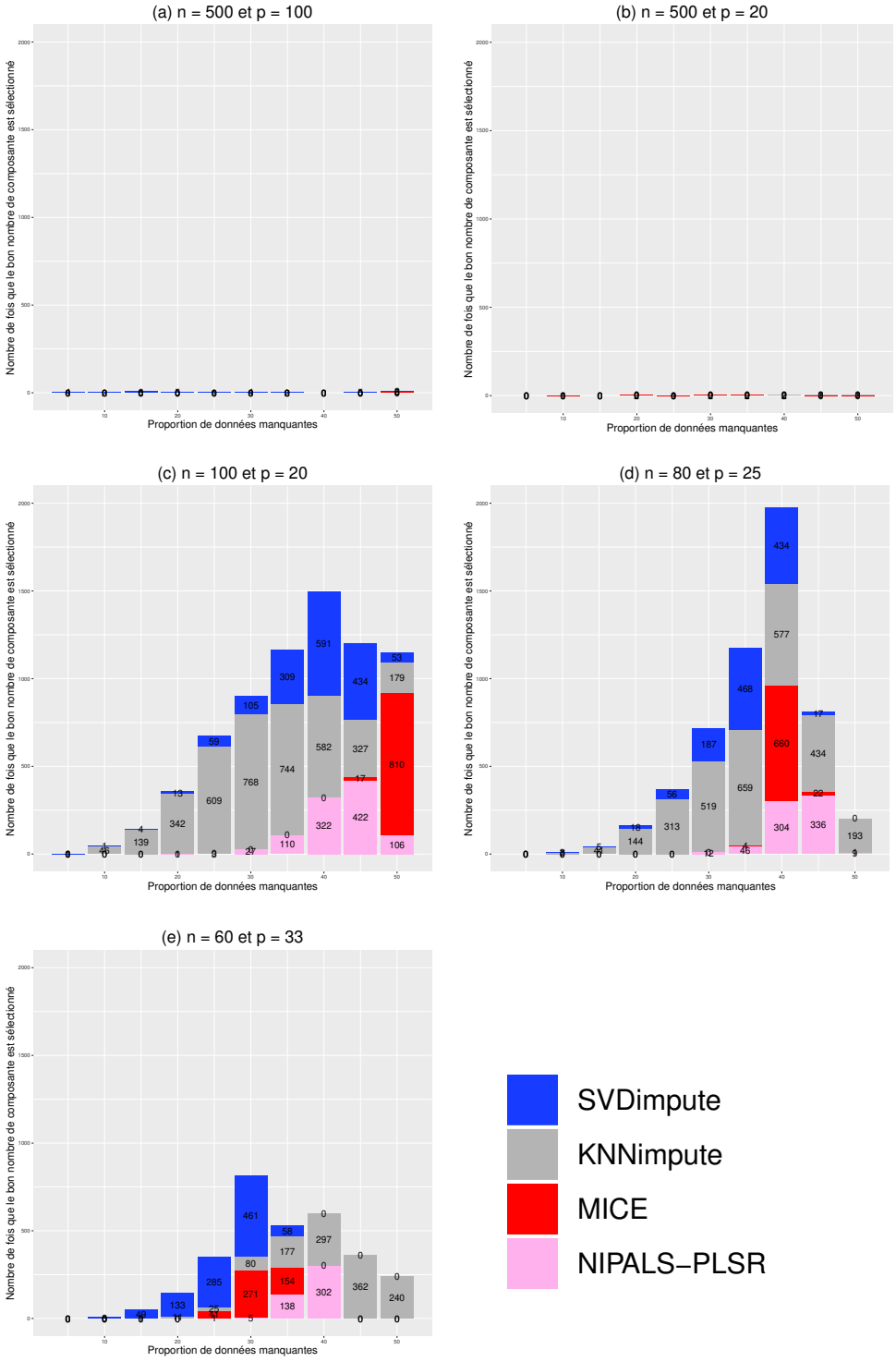


Figure B.29 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

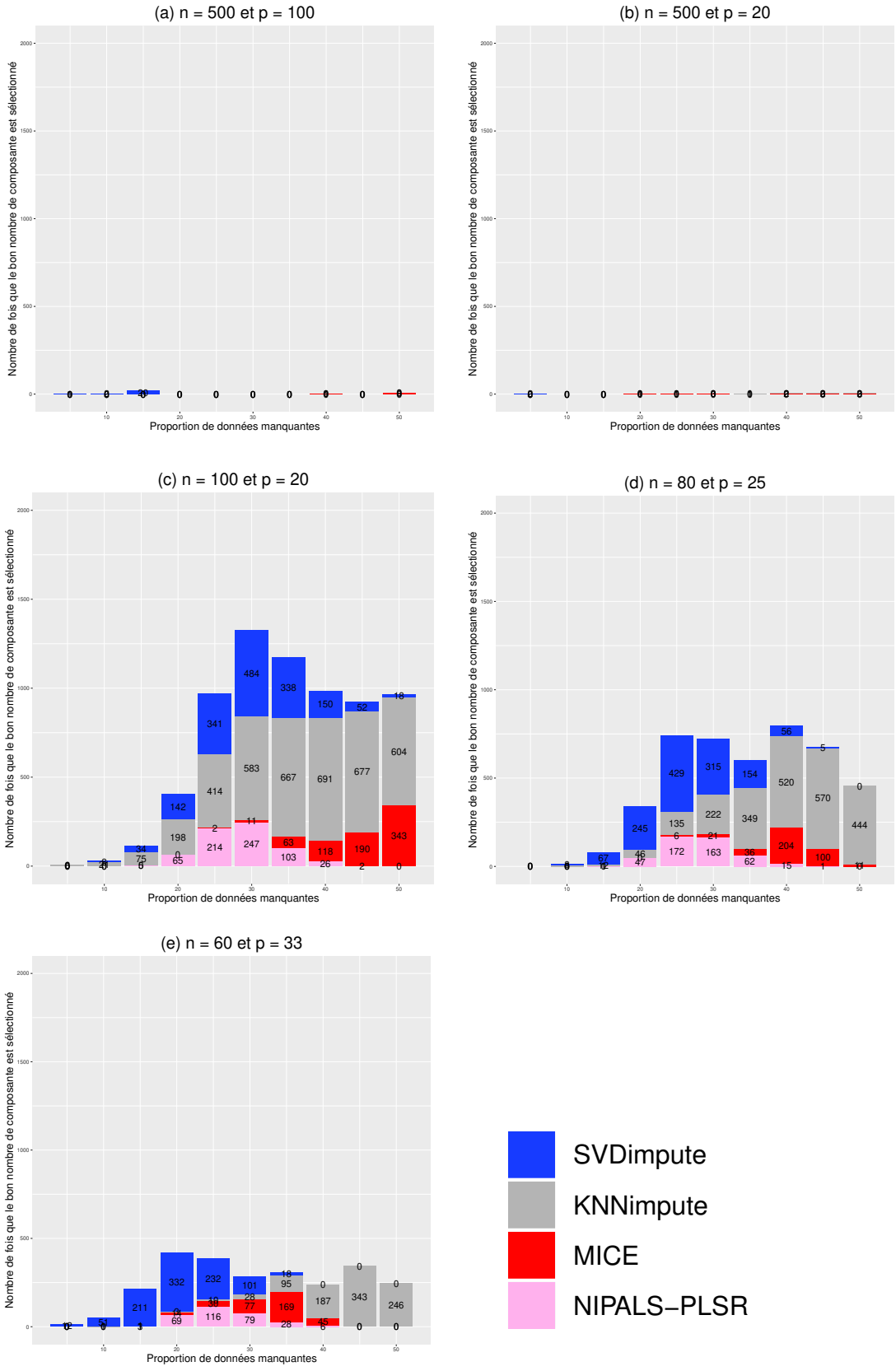


Figure B.30 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC-DoF

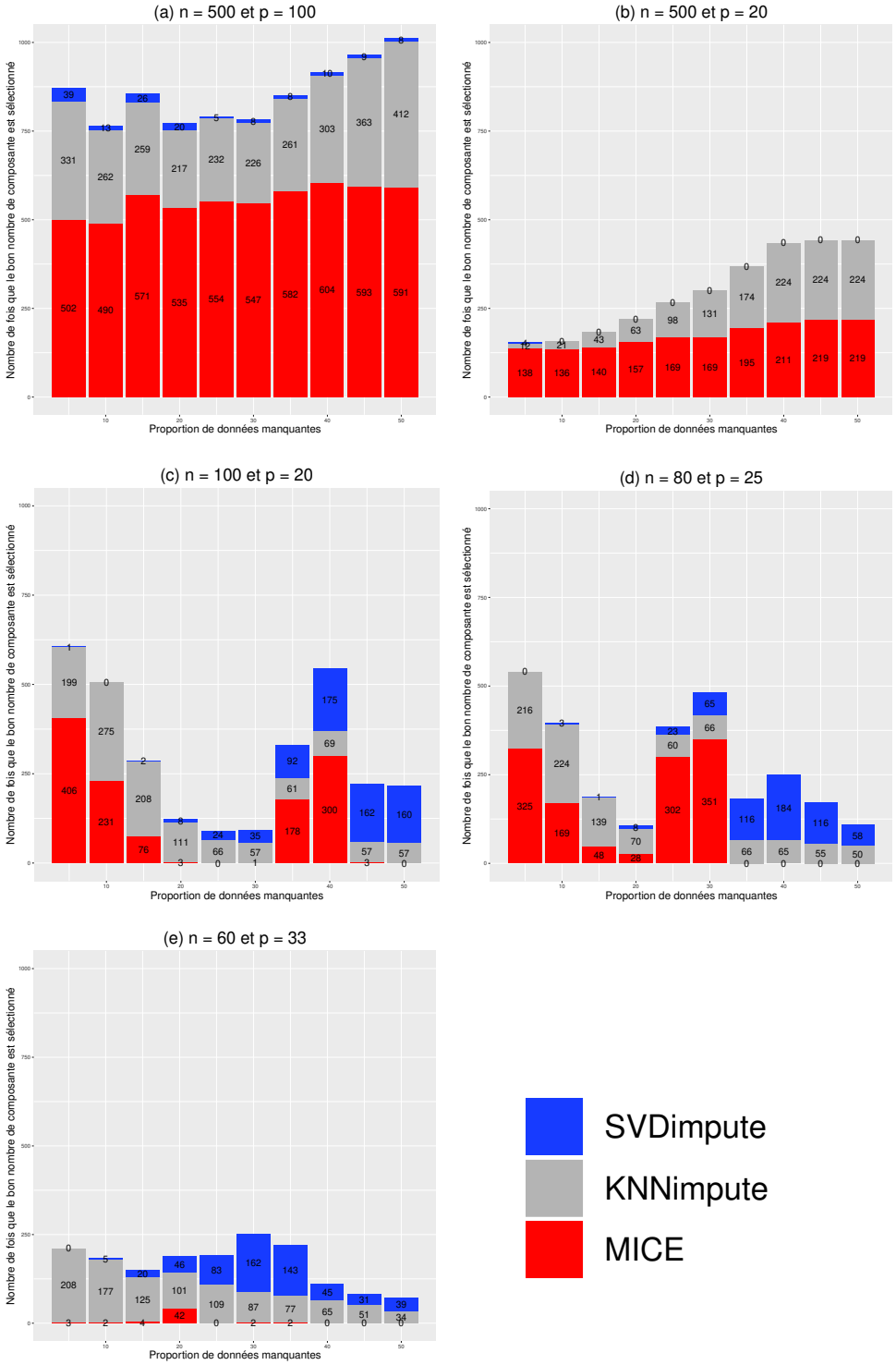


Figure B.31 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

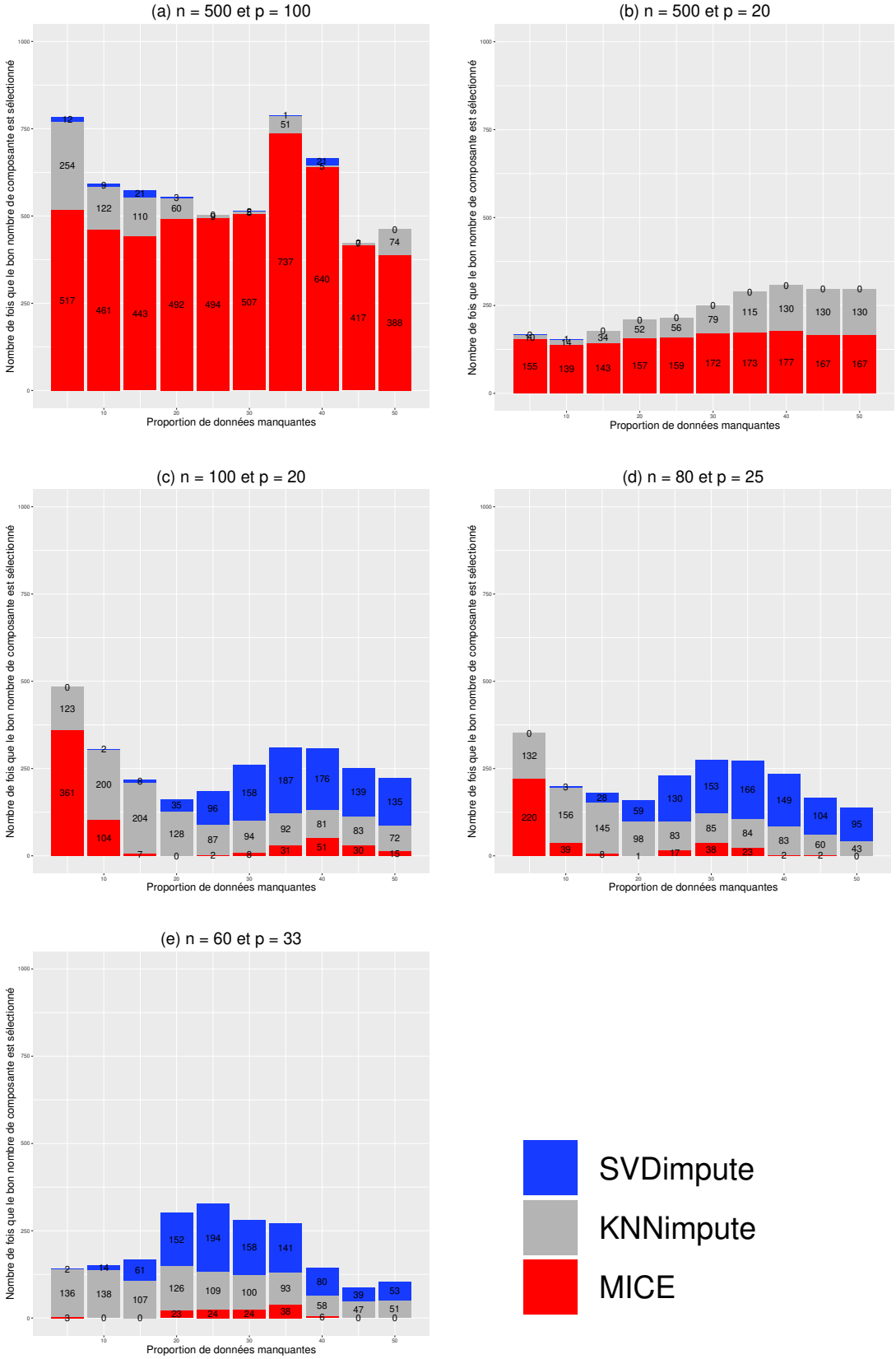


Figure B.32 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC

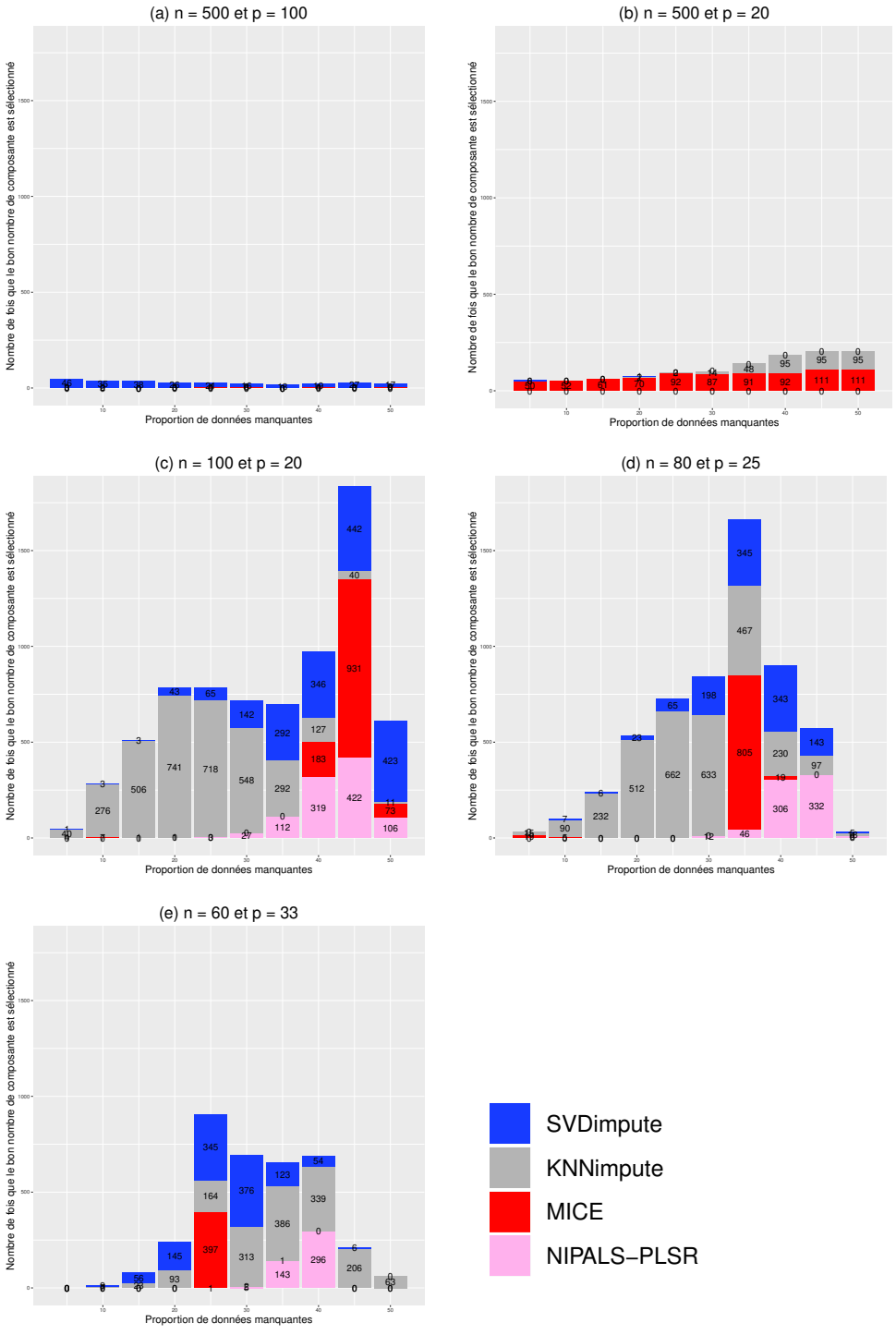


Figure B.33 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

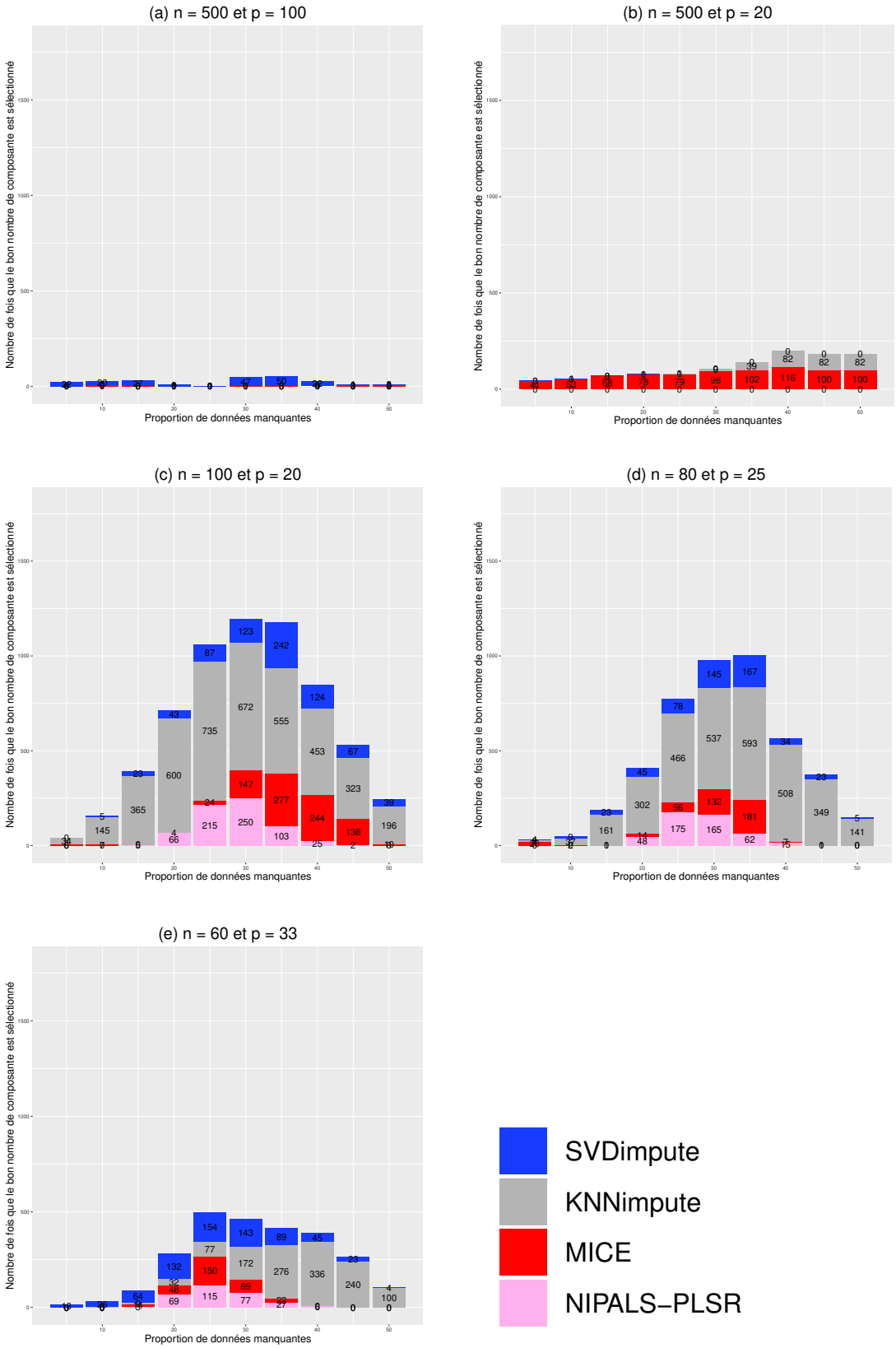


Figure B.34 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC-DoF

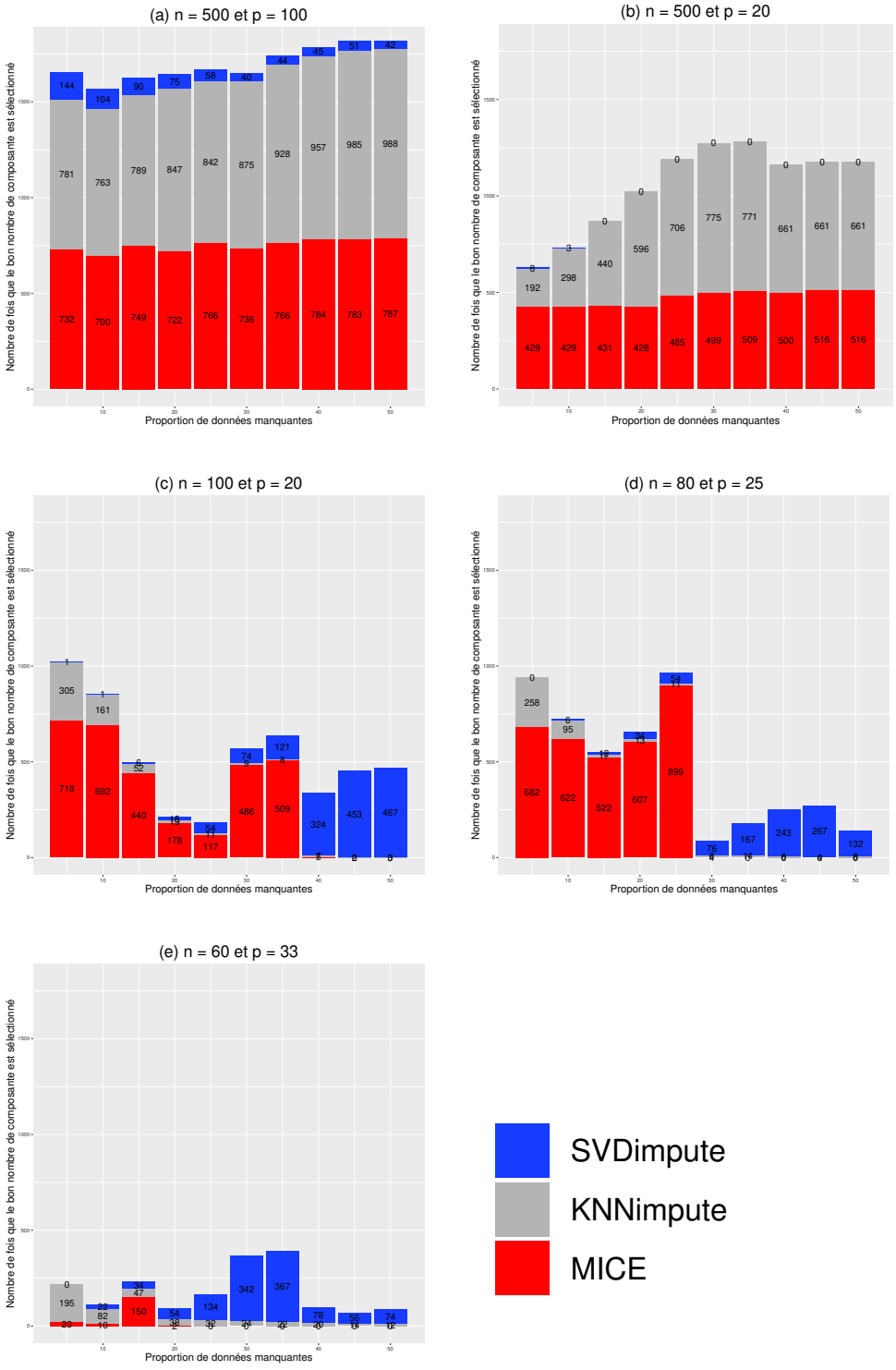


Figure B.35 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

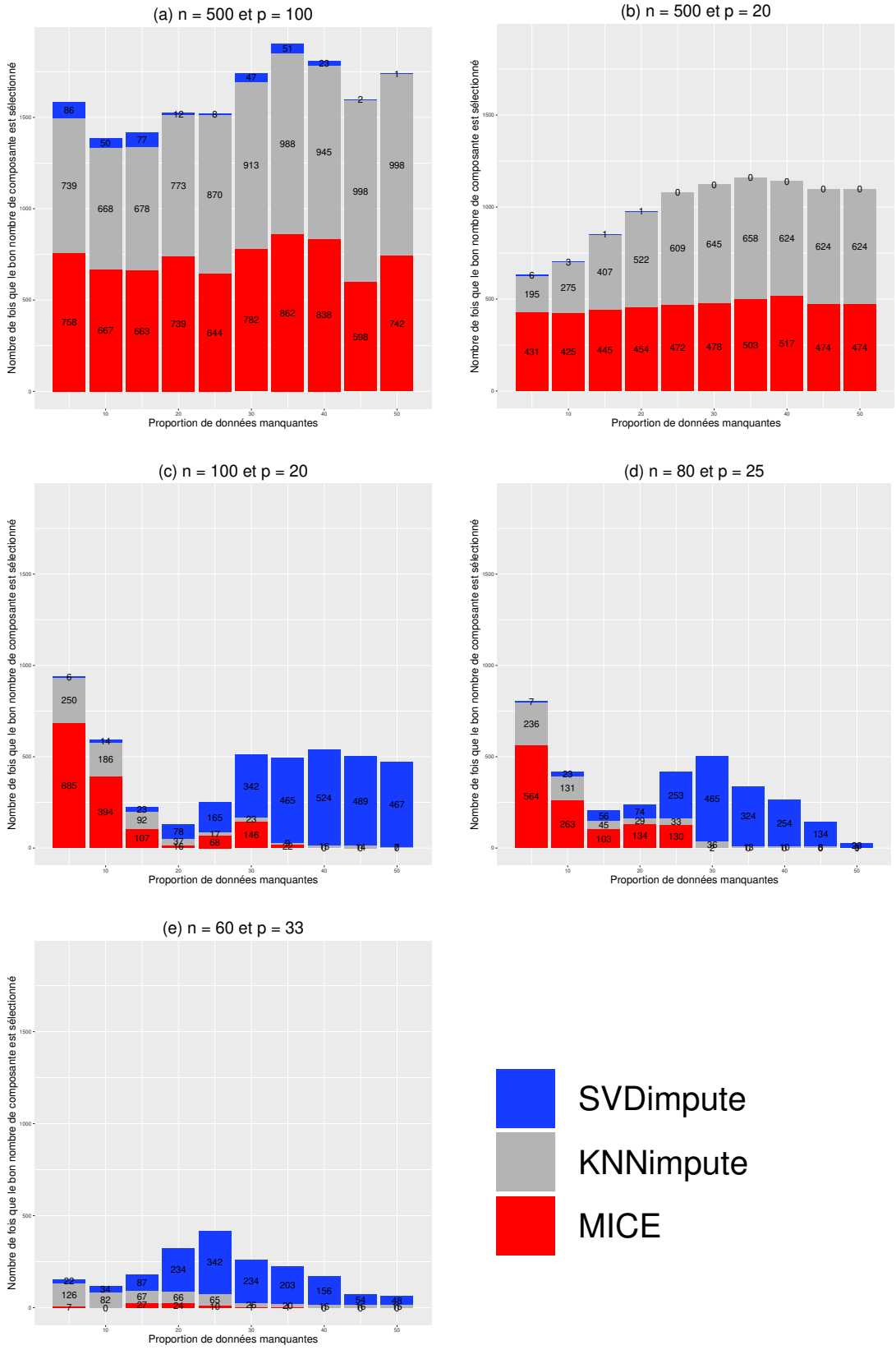


Figure B.36 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

B.2 Données à matrice horizontale

B.2.1 Nombre vrai de composantes = 2

Q^2-LOO

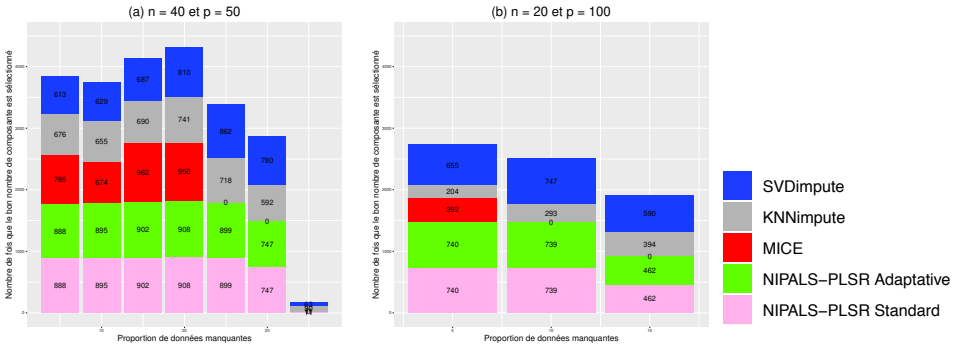


Figure B.37 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

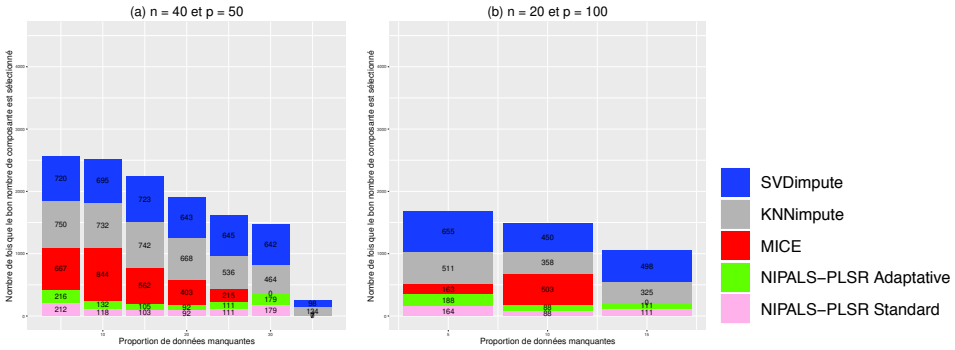


Figure B.38 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

*Q^2-**10**-Fold*

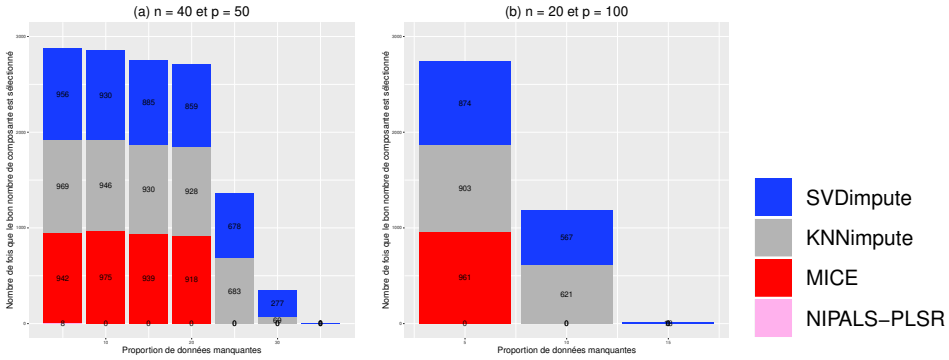


Figure B.39 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

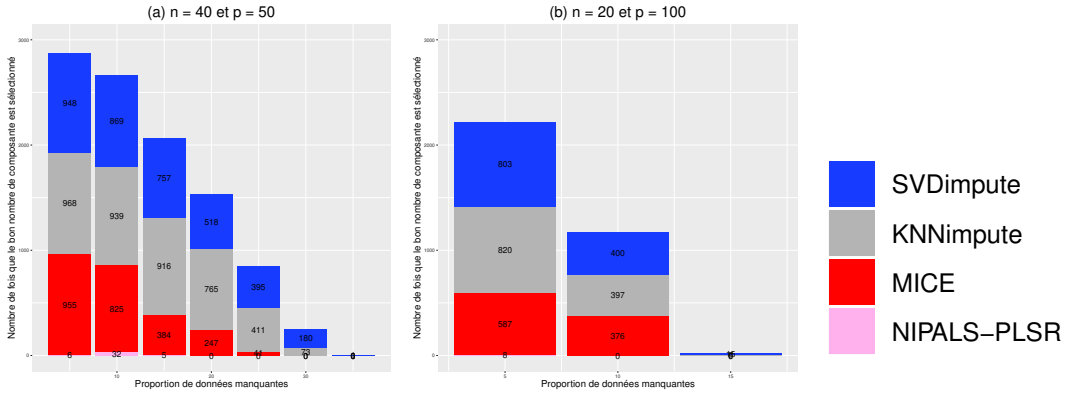


Figure B.40 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC

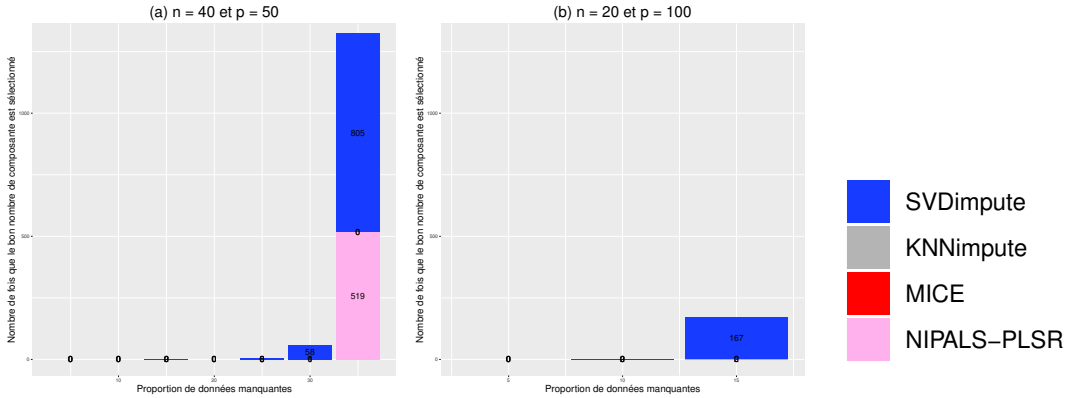


Figure B.41 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

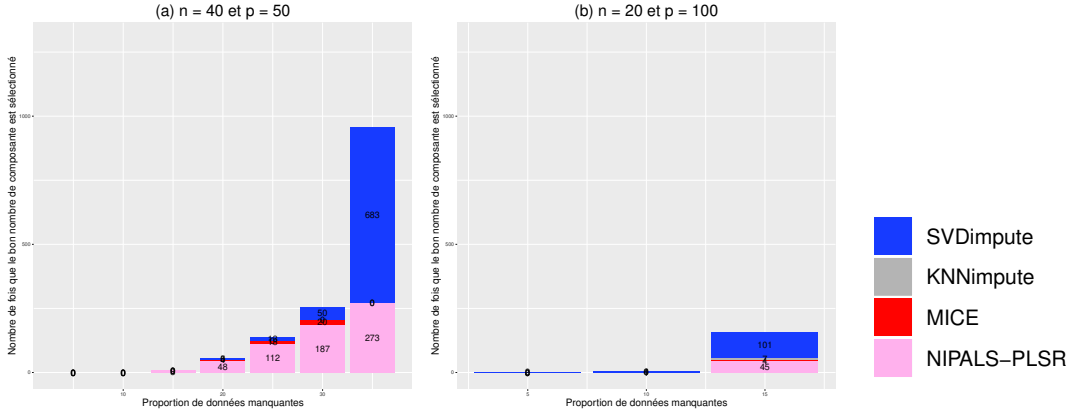


Figure B.42 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC-DoF

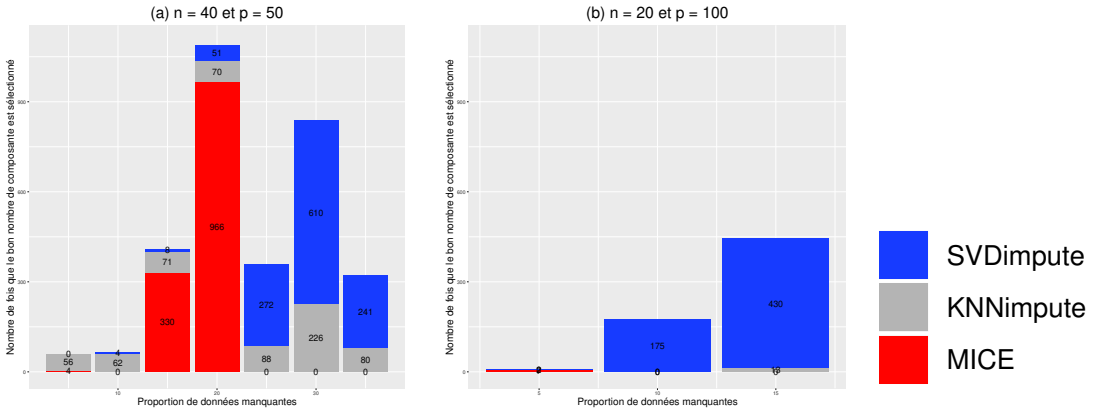


Figure B.43 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

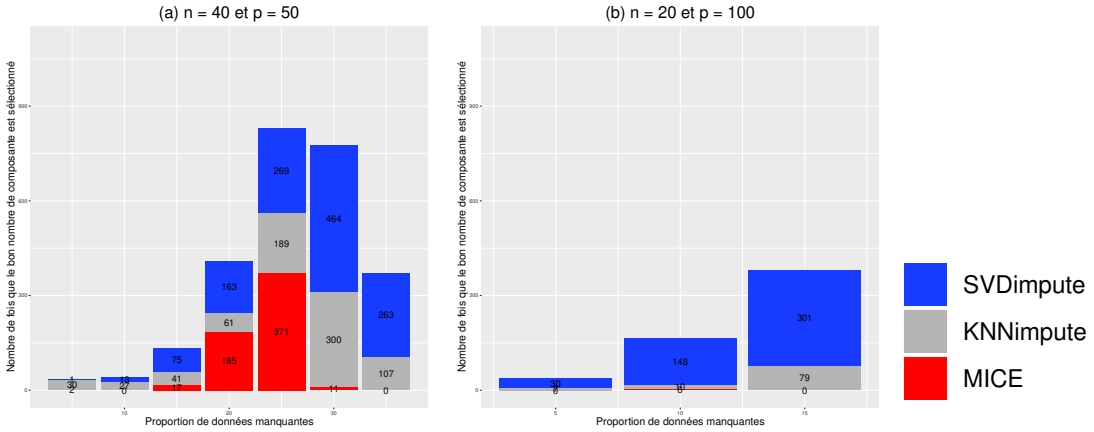


Figure B.44 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC

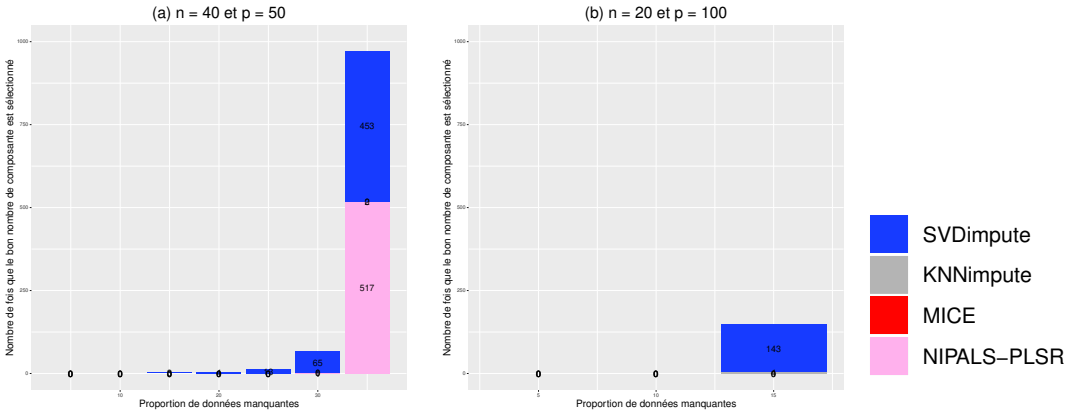


Figure B.45 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

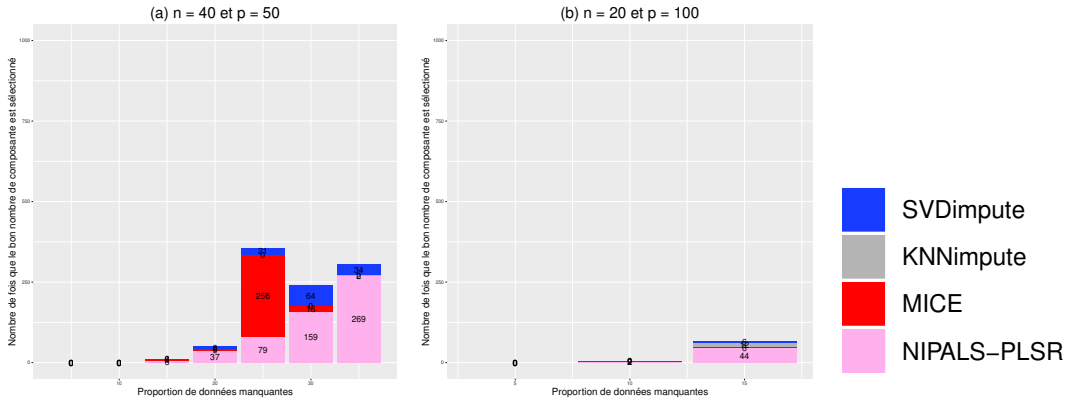


Figure B.46 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC-DoF

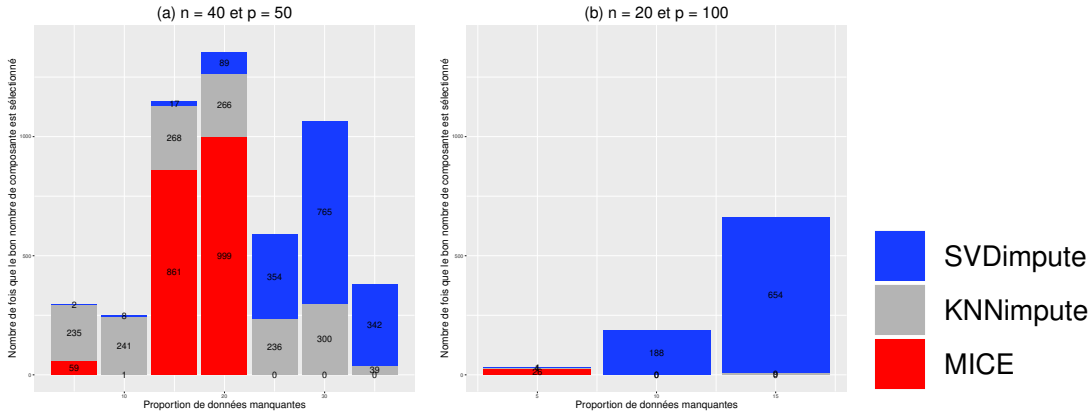


Figure B.47 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MCAR*.

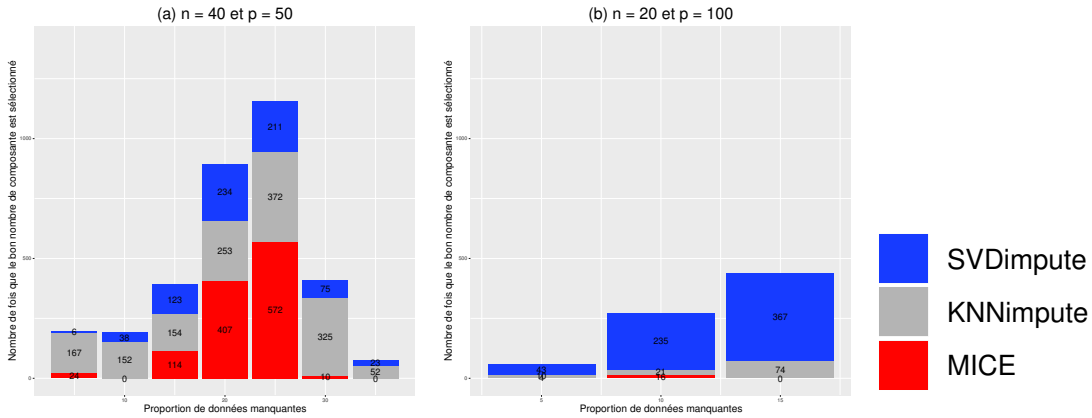


Figure B.48 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 2$ sur 1000 simulations, sous l'hypothèse *MAR*.

B.2.2 Nombre vrai de composantes = 4

Q^2-LOO

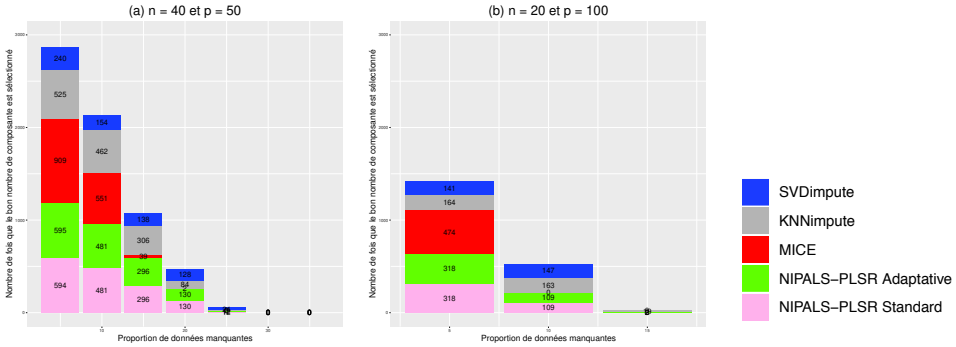


Figure B.49 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

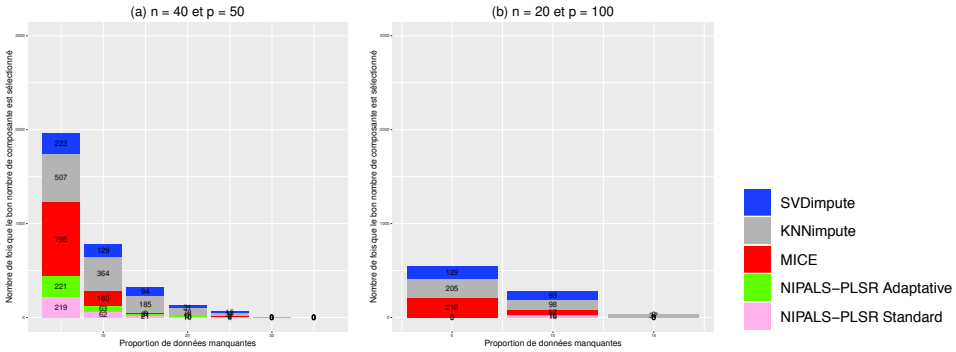


Figure B.50 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

Q^2-10-Fold

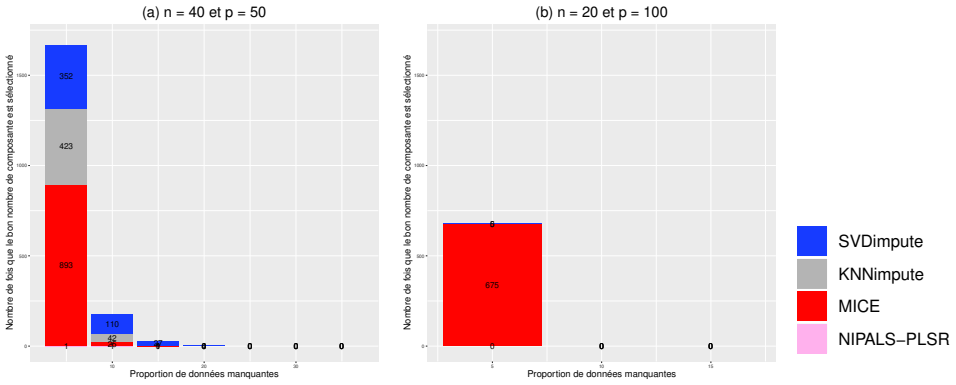


Figure B.51 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

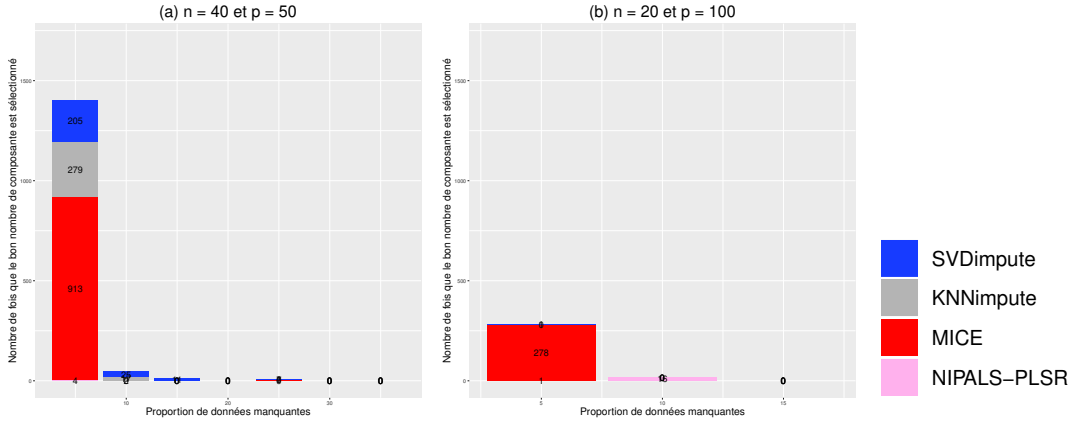


Figure B.52 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC

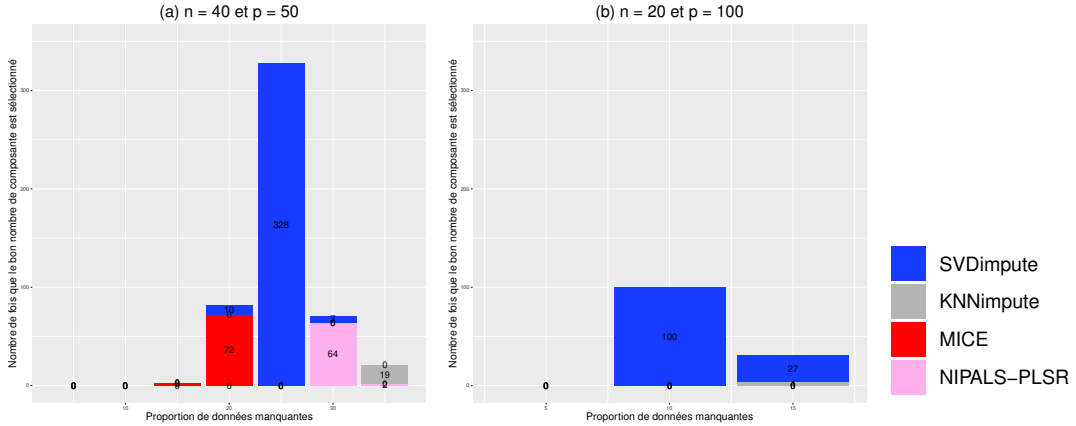


Figure B.53 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

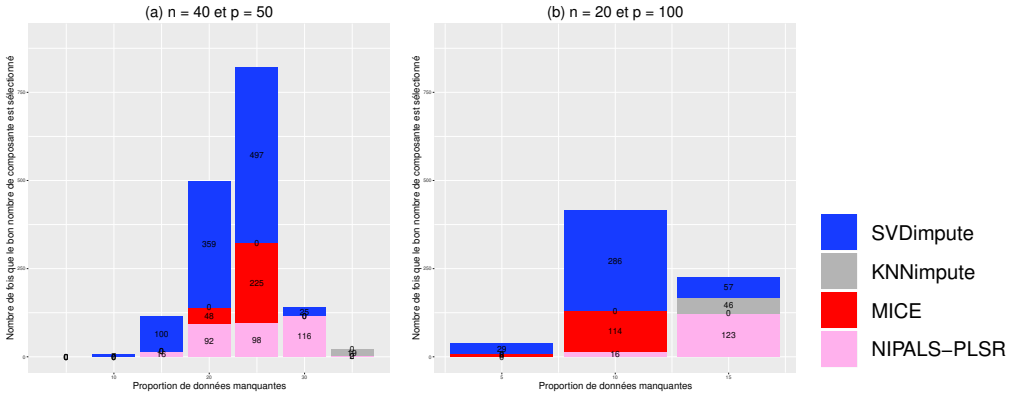


Figure B.54 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC-DoF

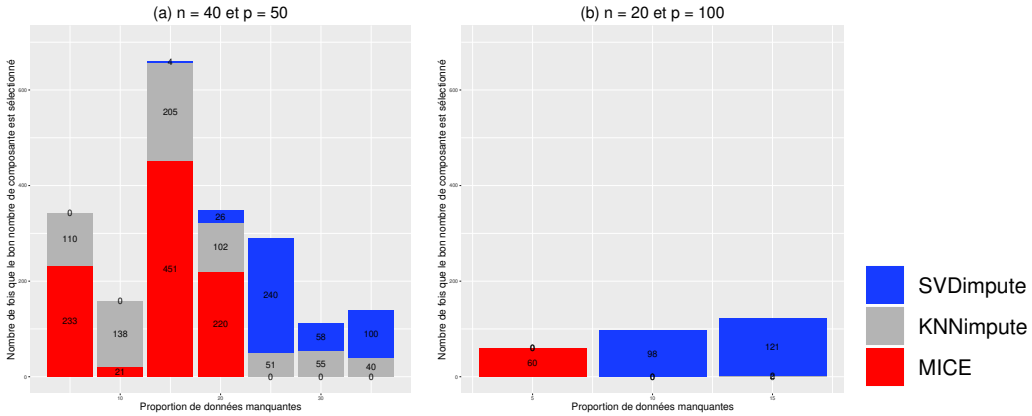


Figure B.55 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

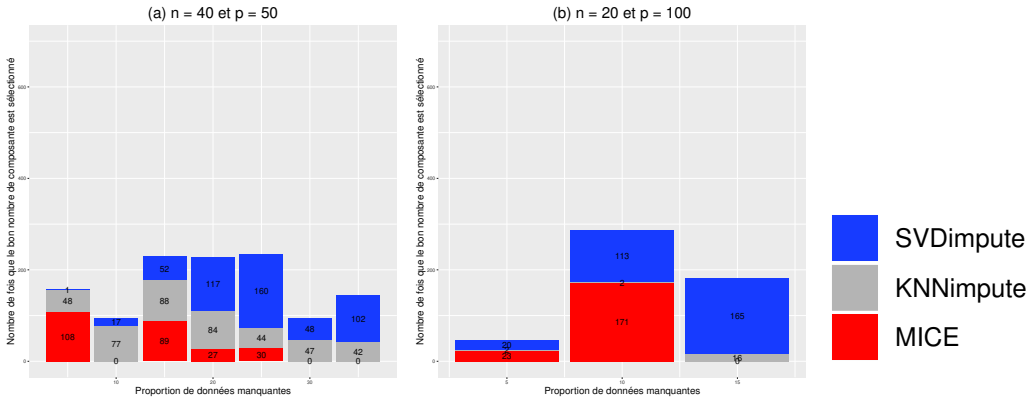


Figure B.56 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC

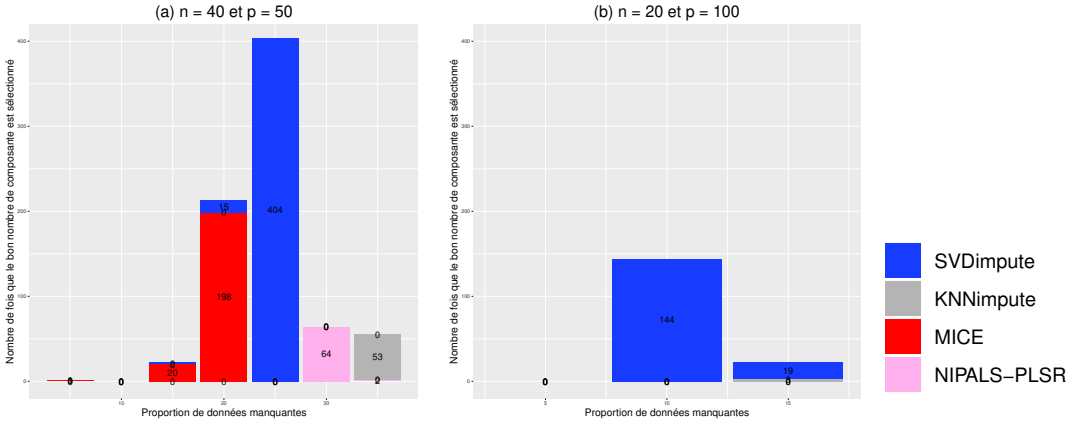


Figure B.57 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

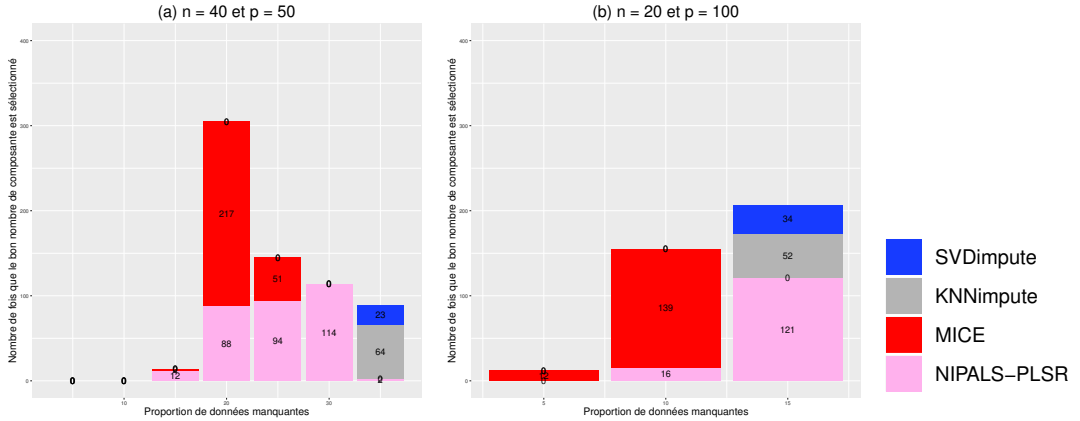


Figure B.58 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC-DoF

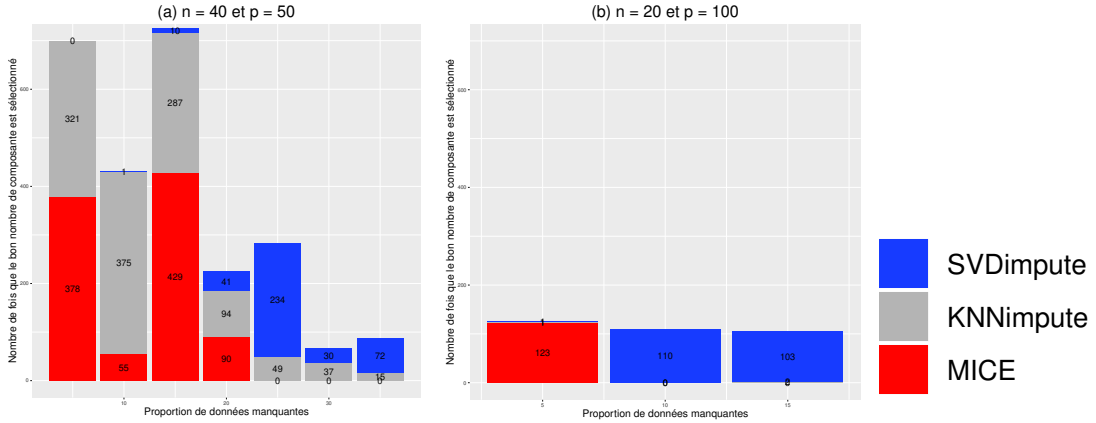


Figure B.59 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MCAR*.

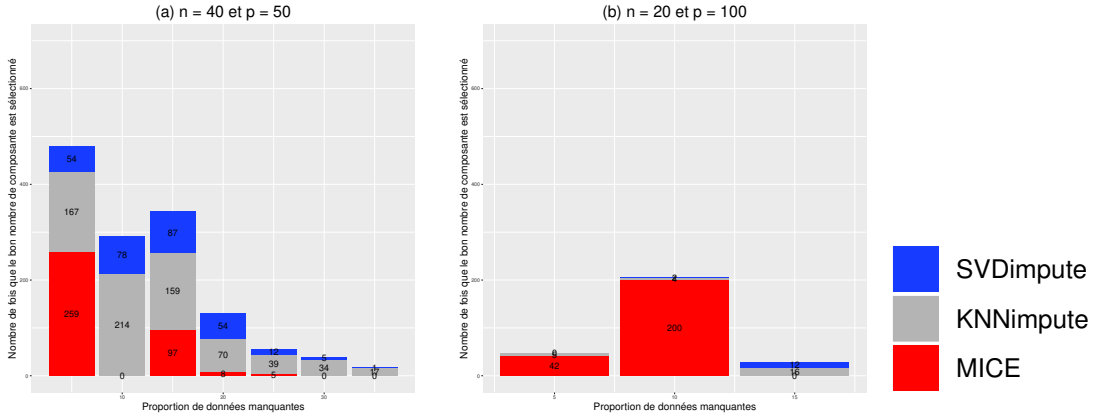


Figure B.60 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 4$ sur 1000 simulations, sous l'hypothèse *MAR*.

B.2.3 Nombre vrai de composantes = 6

Q^2-LOO

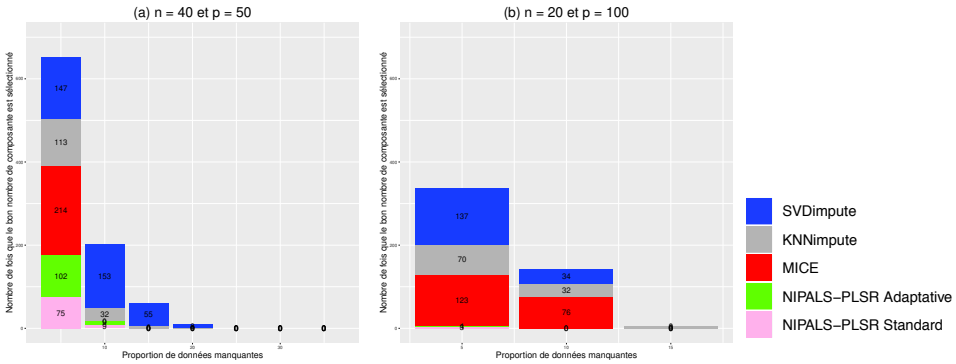


Figure B.61 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

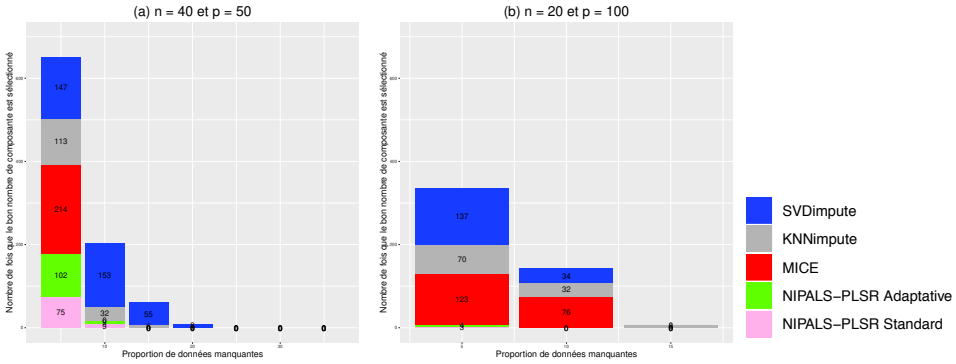


Figure B.62 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-*LOO* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

Q^2-10-Fold

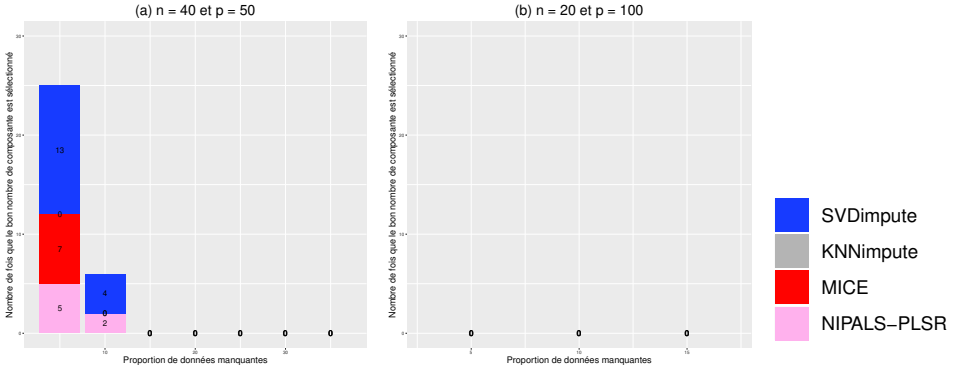


Figure B.63 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

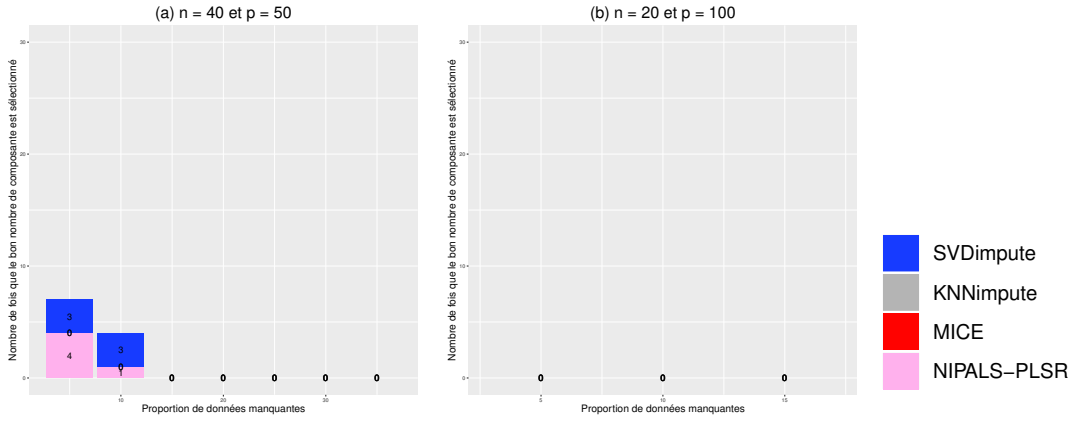


Figure B.64 – Nombre de choix corrects du nombre de composantes avec le critère Q^2-10-*Fold* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC

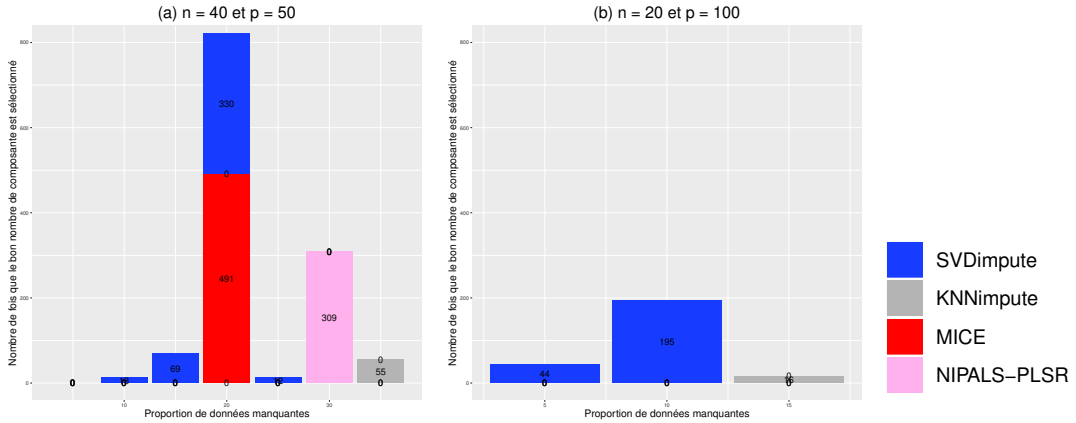


Figure B.65 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

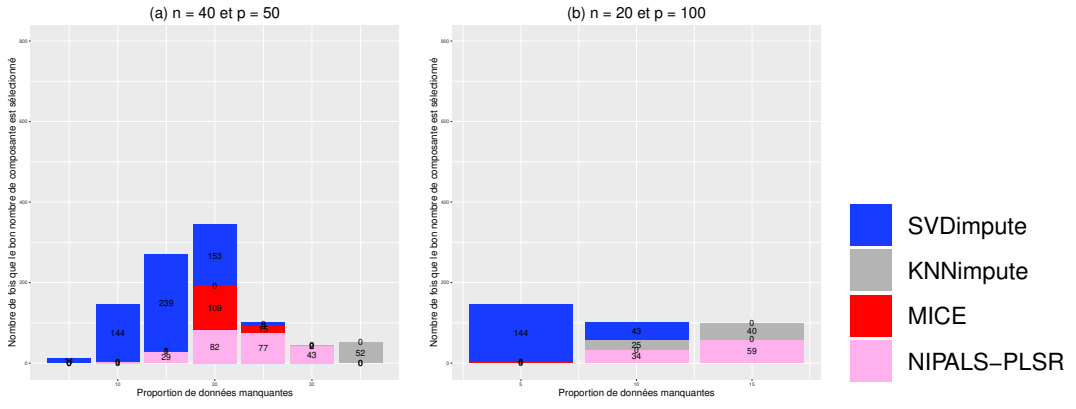


Figure B.66 – Nombre de choix corrects du nombre de composantes avec le critère *AIC* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

AIC-DoF

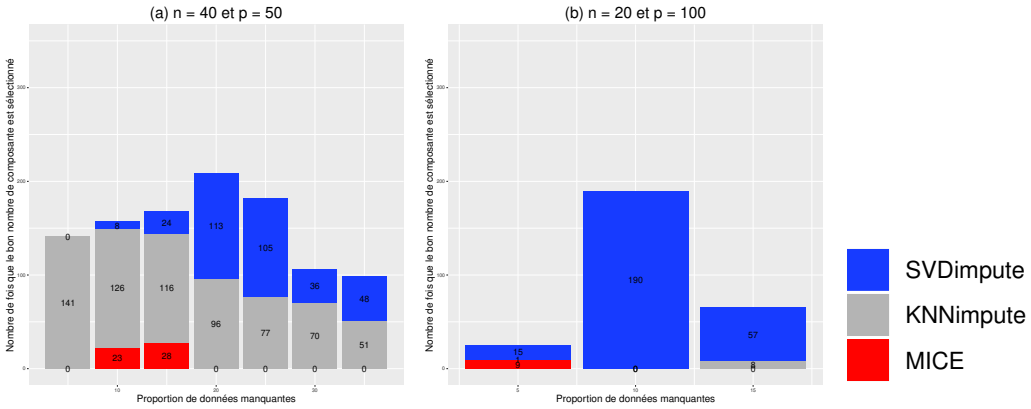


Figure B.67 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

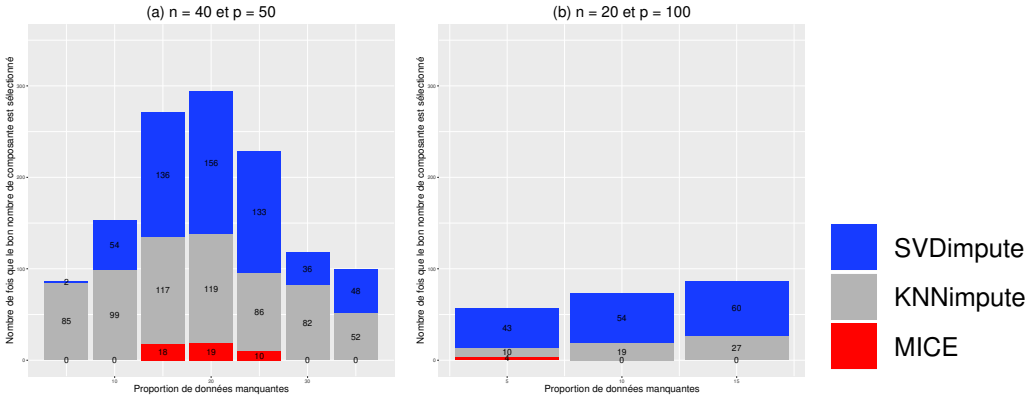


Figure B.68 – Nombre de choix corrects du nombre de composantes avec le critère *AIC-DoF* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC

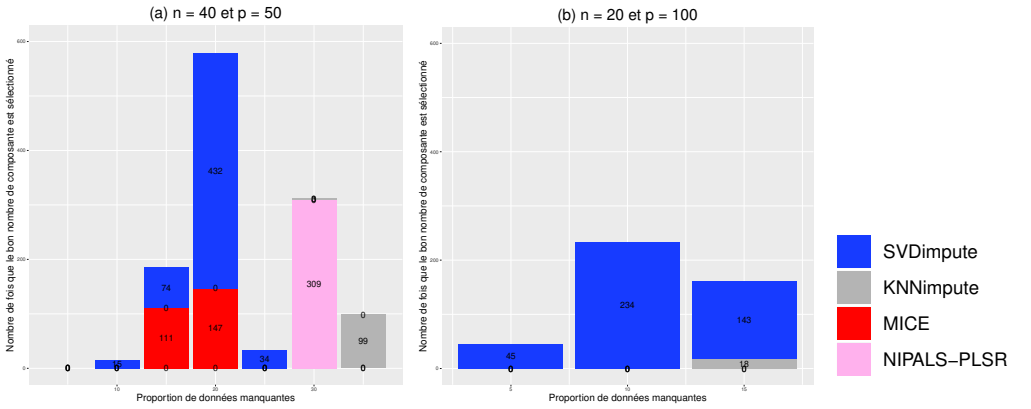


Figure B.69 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

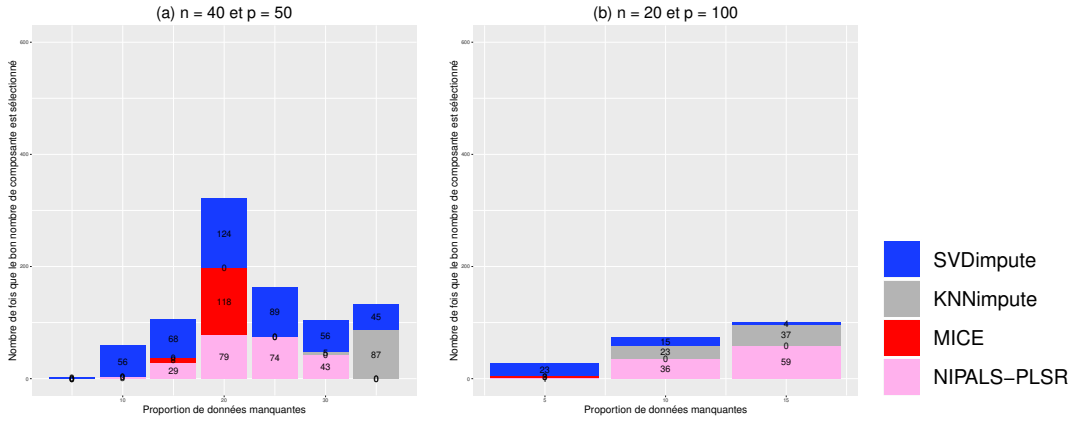


Figure B.70 – Nombre de choix corrects du nombre de composantes avec le critère *BIC* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

BIC-DoF

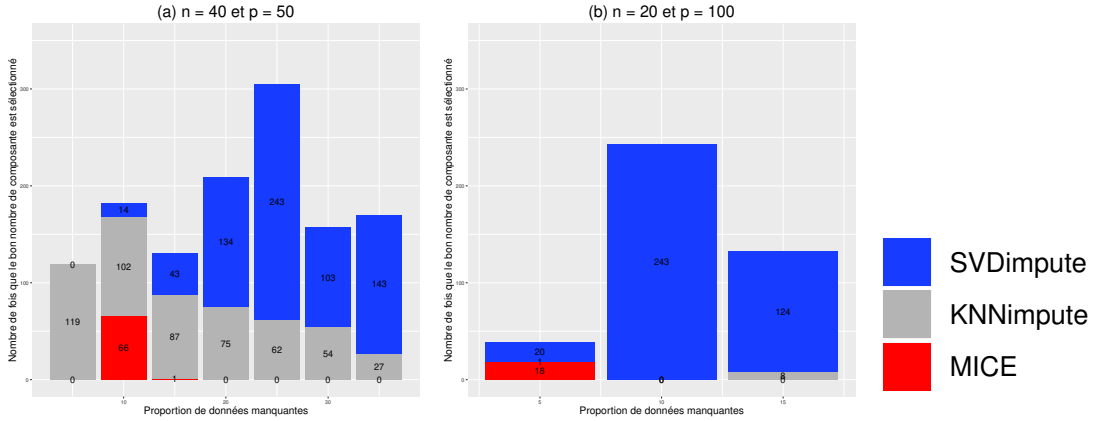


Figure B.71 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MCAR*.

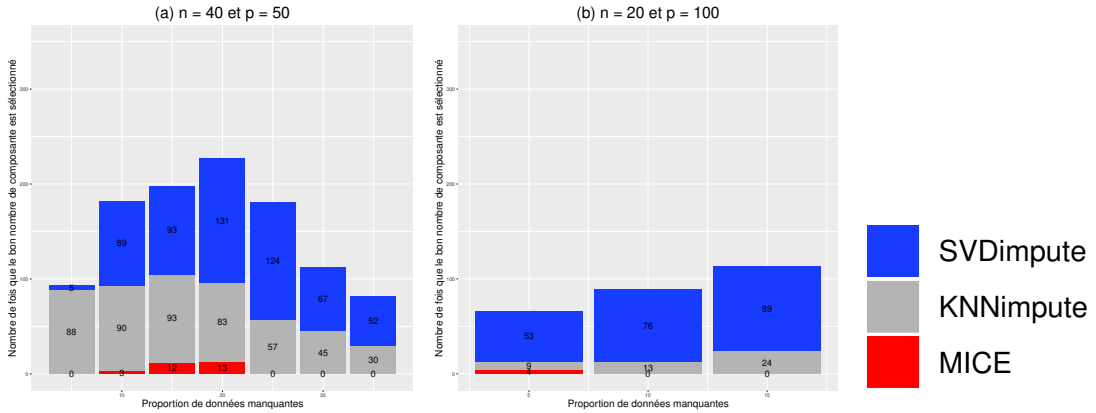


Figure B.72 – Nombre de choix corrects du nombre de composantes avec le critère *BIC-DoF* selon la dimension des données, les méthodes et $t^* = 6$ sur 1000 simulations, sous l'hypothèse *MAR*.

Annexe C

Résumé des temps de calcul

(a) Temps le plus court

t^*	Hypothèses	Dimensions de données						
		$n = 500$ et $p = 100$	$n = 500$ et $p = 20$	$n = 100$ et $p = 20$	$n = 80$ et $p = 25$	$n = 60$ et $p = 33$	$n = 50$ et $p = 40$	$n = 20$ et $p = 100$
2	*MCAR*	*NIPALS-PLS*	*SVDimpute*	*SVDimpute*	*SVDimpute*	*NIPALS-PLS*	*SVDimpute*	*SVDimpute*
	MAR	*NIPALS-PLS*	*SVDimpute*	*SVDimpute*	*NIPALS-PLS*	*NIPALS-PLS*	*NIPALS-PLS*	*NIPALS-PLS*
4	*MCAR*	*NIPALS-PLS*	*SVDimpute*	*SVDimpute*	*SVDimpute*	*NIPALS-PLS*	*NIPALS-PLS*	*SVDimpute*
	MAR	*NIPALS-PLS*	*SVDimpute*	*SVDimpute*	*NIPALS-PLS*	*NIPALS-PLS*	*NIPAL-PLS*	*SVDimpute*
6	*MCAR*	*NIPALS-PLS*	*SVDimpute*	*SVDimpute*	*MICE*	*NIPALS-PLS*	*NIPALS-PLS*	*SVDimpute*
	MAR	*NIPALS-PLS*	*SVDimpute*	*SVDimpute*	*MICE*	*NIPALS-PLS*	*NIPALS-PLS*	*SVDimpute*

(b) Temps le plus long

t^*	Hypothèses	Dimensions de données						
		$n = 500$ et $p = 100$	$n = 500$ et $p = 20$	$n = 100$ et $p = 20$	$n = 80$ et $p = 25$	$n = 60$ et $p = 33$	$n = 50$ et $p = 40$	$n = 20$ et $p = 100$
2	*MCAR*	*SVDimpute*	*NIPALS-PLS*	*MICE*	*MICE*	*MICE*	*KNNimpute*	*KNNimpute*
	MAR	*SVDimpute*	*NIPALS-PLS*	*MICE*	*KNNimpute*	*MICE*	*KNNimpute*	*KNNimpute*
4	*MCAR*	*SVDimpute*	*NIPALS-PLS*	*MICE*	*MICE*	*MICE*	*MICE*	*KNNimpute*
	MAR	*SVDimpute*	*NIPALS-PLS*	*MICE*	*MICE*	*MICE*	*MICE*	*KNNimpute*
6	*MCAR*	*SVDimpute*	*NIPALS-PLS*	*NIPALS-PLS*	*MICE*	*KNNimpute*	*KNNimpute*	*KNNimpute*
	MAR	*SVDimpute*	*NIPALS-PLS*	*MICE*	*MICE*	*KNNimpute*	*KNNimpute*	*KNNimpute*

Annexe D

Poster

D.1 17th applied stochastic models and data analysis

Nous donnons ici le poster présenté dans le cadre de la *17th applied stochastic models and data analysis* qui s'est déroulée à Londres, Angleterre du 6 au 9 juin 2017.

Influence of missing data on the Estimation of the number of Components of a PLS Regression

F. Bertrand[1], T.A. Nengsih[1, 2], M. Maumy-Bertrand[1], and N. Meyer [2,3]

[1] IRMA, LabEx IRMIA, Université de Strasbourg, 7 rue René-Descartes 67084, Strasbourg, France, (E-mail: fbertran@unistra.fr, mmaumy@unistra.fr)

[2] ICube, 300 bd Sébastien Brant, 67400 Illkirch-Graffenstaden, France (E-mail: nengsih@unistra.fr)

[3] GMRC, Pôle de Santé Publique, CHU de Strasbourg, France (E-mail: nmeyer@unistra.fr)

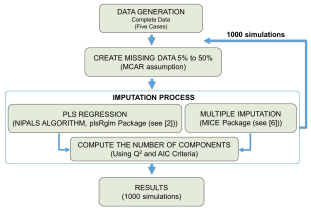
INTRODUCTION

One of the justification of the NIPALS algorithm in PLS regression is that it can be applied on incomplete data sets (see Bastien and Tenenhaus [1]). This property has been acknowledged for long and is frequently used as an argument to preferentially use this algorithm. Though the NIPALS can be considered as the standard algorithm, its reliability to estimate the PLS regression parameters on incomplete data sets has rarely been studied. This literature is sparse on this topic. The sensitivity of NIPALS algorithm to increasing proportion of missing data does not seem to have been given enough attention. Multiple Imputation has become a classical method to estimate regression models on incomplete data. The imputed data are usually generated using Multivariate Imputation by Chained Equation (MICE [6]).

The goal of our simulation study is to analyze the impact of the proportion of missing data (MCAR assumption) on the estimation of the number of components of a PLS regression model. For this, we use two criteria: Q^2 and AIC.

METHODS

1. Les données sont simulées (Li et al, 2002 [3]). Soit n le nombre d'observations, m le nombre de variables et η le nombre réel de composants (mis à 4):
   - n = 20 et m = 100,
   - n = 40 et m = 50,
   - n = 60 et m = 33,
   - n = 80 et m = 25,
   - n = 100 et m = 20.
2. Les données manquantes sont sous l'hypothèse de MCAR et l'hypothèse de MAR .
3. La proportion de données manquantes, noté d, est passée de 5% à 50% par étapes de 5%.
4. Les données manquantes sont imputées en utilisant une imputation multiple avec le paquet MICE.
5. Le nombre de composants est dérivé en utilisant la validation croisée calculée sur les données incomplètes: standard et adaptatif Q2 et AIC (Meyer et al., 2010 [5]). Pour les imputations multiples, le nombre de composants a été calculé par validation croisée sur chaque imputation et en moyenne, ce qui donne un nombre moyen (généralement non entier) de composants.



RESULTS

Table 1. Results are expressed as number of simulations for which the selected components number equals 4 (the true value).

n	d	Q^2	Q^2_{cv}	$Q^2_{standard}$	$Q^2_{adaptive}$	AIC	AIC_{dof}	$AIC_{standard}$	$AIC_{adaptive}$
20	5	499	891	590	590	0	0	0	0
	10	474	908	440	440	0	0	0	0
	15	498	917	63	63	0	0	0	0
40	5	1000	994	781	781	0	329	0	0
	10	999	996	720	720	0	333	0	0
	15	1000	996	633	633	0	341	0	0
	20	1000	999	540	540	0	332	0	0
	25	998	994	292	292	0	306	0	0
	30	1000	997	32	32	0	313	237	237
	35	999	995	0	0	0	316	172	172
60	5	1000	1000	830	861	0	512	0	0
	10	1000	1000	771	772	0	527	0	0
	15	1000	1000	760	760	0	522	0	0
	20	1000	1000	648	648	0	497	0	0
	25	1000	1000	592	592	0	505	0	0
	30	1000	1000	398	398	0	504	9	0
	35	1000	1000	268	268	0	506	70	70
	40	1000	999	85	85	0	509	664	664
	45	999	999	4	4	0	491	712	712
	50	1000	1000	0	0	0	495	5	5
80	5	1000	1000	941	986	40	664	0	0
	10	1000	1000	940	952	43	639	0	0
	15	1000	1000	913	913	44	673	0	0
	20	1000	1000	884	884	47	656	0	0
	25	1000	1000	808	808	40	640	0	0
	30	1000	1000	727	727	41	648	1	1
	35	1000	1000	609	609	40	634	17	17
	40	1000	999	453	453	39	622	176	176
	45	1000	1000	8	8	35	649	148	148
	50	1000	1000	1	1	44	642	75	75
100	5	1000	1000	931	998	357	718	0	0
	10	1000	1000	928	964	364	714	0	0
	15	1000	1000	909	926	388	709	0	0
	20	1000	1000	878	882	404	728	0	0
	25	1000	1000	864	864	373	692	1	1
	30	1000	1000	798	799	397	701	6	6
	35	1000	1000	746	746	384	720	34	34
	40	1000	1000	648	648	403	712	156	156
	45	1000	1000	498	498	387	712	469	469
	50	1000	1000	342	342	354	717	862	862

Explanation Table 1:

- The number of components selected using the Q^2 and Q^2_{cv} criteria is on average equal to 95%.
- The $Q^2_{standard}$ and $Q^2_{adaptive}$ select the correct number of components less frequently than the Q^2 and the Q^2_{cv}.
- The AIC seems to identify the correct number of components only in the last case(n= 100) but in 35% of time.
- The AIC_{dof} (see [4]) works better than the AIC's.
- The indices derived from the AIC ($AIC_{standard}$ and $AIC_{adaptive}$) are sometimes more efficient than the AIC. They identify the correct number of components only in a minority of situations, even if their performances are better for the last case (n= 100).

Explanation Table 2:

- The number of components selected by Q^2 is larger for increasing sample size.
- The Q^2 and Q^2_{cv} show a more consistent decreasing pattern for an increasing proportion of missing data.
- The number of components selected by the AIC, AIC_{dof} and AIC_{naive} are almost twice the true number of components.

Table 2. Results for Multiple Imputations are given as "modal value of modal of components over 5 imputations / number of simulations for that modal value".

Example for n = 20, d = 5%, Q^2_{cv} (fourth column), "2/601" means that 2 components were the value which appeared most often over the 5 imputations and that for this modal value of selected components, for a given set of 1000 simulations, this modal value was observed at least 601 times.

n	d	Q^2	Q^2_{cv}	AIC	AIC_{dof}	AIC_{naive}
20	5	2/245	2/601	8/1000	8/998	8/1000
40	5	4/809	4/898	8/987	8/548	8/987
60	5	4/693	4/750	8/752	8/349	8/742
	10	4/676	4/739	8/850	8/375	8/842
	15	4/631	4/774	8/976	8/431	8/974
	20	4/689	4/460	8/968	5/270	8/964
	25	2/694	1/490	8/477	3/405	8/461
80	5	4/912	4/923	6/590	4/547	6/594
	10	4/892	4/907	6/471	4/501	6/473
	15	4/850	4/894	7/442	4/401	7/433
	20	4/771	4/862	6/647	8/390	8/646
	25	4/753	4/838	4/853	8/507	8/848
	30	4/674	4/502	8/866	8/328	8/865
	35	3/707	3/520	8/549	4/433	8/543
	40	2/793	2/706	6/311	3/543	6/312
	45	1/721	1/942	4/277	2/672	4/281
	50	1/956	1/926	4/338	1/489	4/339
100	5	4/987	4/990	6/643	4/664	6/640
	10	4/977	4/987	6/650	4/629	6/643
	15	4/943	4/973	6/571	4/492	6/570
	20	4/866	4/947	7/355	4/322	7/357
	25	4/816	4/916	8/541	8/475	8/576
	30	4/815	4/890	8/745	8/578	8/740
	35	4/834	4/804	8/816	8/537	8/812
	40	4/702	4/478	8/657	8/268	8/649
	45	3/589	3/659	8/405	5/289	8/403
	50	3/475	2/758	6/341	4/574	6/342

CONCLUSION

Our simulation study shows that:

- The correct number of components of PLS regression model is difficult to determinate, especially for small sample size and when the proportion of missing data is larger than 30%.
- The Q^2 criterion and its derived show the best performance whatever the missing data proportion.
- The performances the Q^2 and AIC increase with an increasing of the sample size and get worse when the proportion of missing data increase.

REFERENCES

1. Bastien P. and Tenenhaus M., PLS regression and multiple imputations, Proceedings of the PLS'03 International Symposium, Vilares M et al. editors CISIA Paris, pp. 497-498, (2003).
2. Bertrand F., Meyer N. and Maumy-Bertrand M., Partial least squares regression for generalized linear models, (Book of abstracts, User2014! Los Angeles, p. 150, (2014).
3. Li B., Morris J. and Martin EB., Model selection for partial least squares regression. Chemometrics and Intelligent Laboratory Systems 64: 79-89, (2002).
4. Krämer N. and Sugiyama N., The degrees of freedom of partial least squares regression, Journal of the American Statistical Association 106(494): 697-705, (2011).
5. Meyer N., Maumy-Bertrand M. and Bertrand F., Comparaison de variantes de régressions logistiques PLS et de régression PLS sur variables qualitatives : application aux données d'alllotypage, JSFdS 150(2): 1-18, (2010).
6. Van Buuren S. and Groothuis-Oudshoorn K., mice: Multivariate Imputation by Chained Equations in R, Journal of Statistical Software 45(3): 1-67, (2011).

Figure D.1 – 17th applied stochastic models and data analysis.

D.2 Doctoral School Days

Nous donnons ici le poster présenté dans le cadre de la *Doctoral School Days* qui s'est déroulée au Collège Doctoral Européenne Unistra à Strasbourg, France du 8 au 9 mars 2018.

Determining the Number of Components for a PLS Regression on Incomplete Data for MCAR Assumption

T.A. Nengsih[1,3], N. Meyer [1,2], F. Bertrand[3], and M. Maumy-Bertrand[3]

[1] ICube, 300 bd Sébastien Brant, 67400 Illkirch-Graffenstaden, France, (E-mail: nengsih@unistra.fr)

[2] GMRC, Pôle de Santé Publique, CHU de Strasbourg, France (E-mail: nmeyer@unistra.fr)

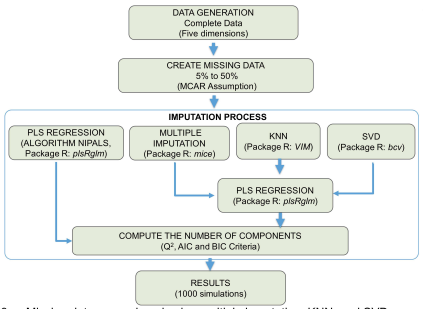
[3] IRMA, LabEx IRMIA, Université de Strasbourg, 7 rue René-Descartes 67084 Strasbourg, France, (E-mail: fbertran@unistra.fr, mmaumy@unistra.fr)

INTRODUCTION

Missing data is known to be a concern for the applied researcher. Several methods have been developed for handling incomplete data. Imputation is the process of substituting missing data before estimating the relevant model parameters. PLS regression is a multivariate model for which two algorithms (SIMPLS or NIPALS) can be used to provide its parameters estimates. The NIPALS algorithm has the interesting property of being able to provide estimates on incomplete data. Furthermore, selection of the number of components to build a representative model in PLS regression is an important problem. Fitting the number of components of a PLS regression on incomplete data set leads to the problem of model validation, which is generally done using cross-validation. Determination of the number of components relies on several different criteria such as the Q^2 criterion, the Akaike Information Criterion (AIC), or the Bayesian Information Criteria (BIC).

We compared the criteria for selection of the number of components of a PLS regression according to PLS regression with NIPALS algorithm (NIPALS-PLSR) on incomplete data and PLS regression on imputed data set using three methods of imputation: multiple imputation by chained equations (MICE), K-nearest neighbors imputation (KNN) and a singular value decomposition based method (SVD). The criteria compared were Q^2-LOO, Q^2 10-fold, AIC, AIC-DoF, BIC and BIC-DoF. The goal of our simulation study was to analyze the impact of the missing data proportion (under missing completely at random (MCAR) assumption) on the estimation of the number of components of a PLS regression.

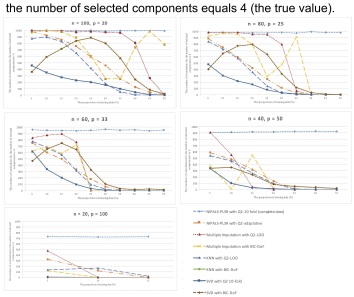
METHODS



1. Data are simulated according to Li *et al.* (2002) which the true number of components is set to 4, n (number of observations) and p (number of variables) with each of the following five dimensions sets:
   - n = 100 and p = 20,
   - n = 80 and p = 25,
   - n = 60 and p = 33,
   - n = 40 and p = 50,
   - n = 20 and p = 100.
2. Missing data are created under MCAR assumption with an increasing proportion of missing data from 5% to 50% by 5% step.
3. Missing data are replaced using multiple imputation, KNN, and SVD.
4. The number of components is computed using LOO (Leave One Out) cross-validation and 10-fold cross-validation computed on the incomplete data according to each of two methods: standard and adaptative (selects the prediction method accordingly to the completeness of the row) (Bertrand *et al.*, 2015). Under multiple imputation, the number of components has been computed by cross-validation as the modal value of the computed number of components across all m imputations where m is equal to 100 x the proportion of missing data (White *et al.*, 2001).
5. Computing the number of components based on the smallest value of Q^2, AIC, and BIC across all number of components.
6. For each combination of the proportion of missing data and matrix dimensions, 1000 replications have been drawn

RESULTS

Figure. The evaluation of the Q^2, the AIC, and the BIC of the four methods (NIPALS-PLSR, multiple imputation, KNN and SVD) and the results are expressed as the number of simulations for which the number of selected components equals 4 (the true value).



Explanation:

After analyzing the four methods used, the selected criteria with the number of components of a PLS regression which are close to the correct number of components over 1000 simulations are:

1. NIPALS-PLSR on complete data: Q^2 10-fold.
2. NIPALS-PLSR on incomplete data: Q^2 adaptative.
3. Multiple imputation: Q^2-LOO and BIC-DoF.
4. KNN: Q^2-LOO and BIC-DoF.
5. SVD: Q^2 10-fold and BIC-DoF.

On vertical matrices (n > p):

- The number of selected components by multiple imputation with the Q^2-LOO criterion is much closer to the correct number of components when the proportion of missing data are small.
- NIPALS-PLSR with the Q^2 adaptative criterion and the KNN with the Q^2-LOO criterion give a slightly better estimation of the true number of components, even if the number of selected components are still far from the true component. The performances increase with increasing sample size and decrease with increasing proportion of missing data.
- The number of selected components using the Q^2-LOO and Q^2 10-fold shows a more consistent decreasing pattern for an increasing proportion of missing data.
- The BIC-DoF criterion, either on the KNN or SVD imputation selects the correct number of components less unstable. Their performances increase and then decrease for an increasing proportion of missing data.
- The running time of multiple imputation was about 5 times slower than NIPALS-PLSR on incomplete data for n= 20 and 25 times slower than NIPALS-PLSR on incomplete data for n= 100.

On the horizontal matrices (n < p):

- For the smaller sample sizes, it was more difficult to reach convergence of the algorithm, especially on the MICE algorithm.
- A smaller proportion of missing data can make it difficult to estimate the correct number of components in a PLS.

Table. The average of running time (seconds) were calculated over 1000 simulations using 6 * 2793 MHz with 518GB RAM

Dimension	NIPALS-PLSR on complete data	NIPALS-PLSR standard on incomplete data	NIPALS-PLSR adaptative on incomplete data	Multiple imputation	KNN	SVD
n = 100, p = 20	0,44	1,21	1,13	28,77	1,08	0,80
n = 80, p = 25	0,40	1,08	1,03	33,99	1,23	0,91
n = 60, p = 33	0,37	0,90	0,87	18,37	1,55	1,10
n = 40, p = 50	0,38	0,92	0,89	10,53	2,50	1,29
n = 20, p = 100	0,52	1,07	1,04	4,92	7,34	0,84

CONCLUSION

Our simulation study shows that:

- Whatever the criterion used, the correct number of components of a PLS regression is difficult to determine, especially for small sample size and when the proportion of missing data is larger than 30%.
- The Q^2-LOO and Q^2 10-fold show the best performance whatever the methods of imputations. The performances increase with the sample size and decrease with increasing proportion of missing data.
- The number of selected components by the AIC, the AIC-DoF and the BIC are almost twice as large as the true number of components.
- The performance of the BIC-DoF criterion is less unstable or inconsistent, which sometimes decreases or increases.

REFERENCES

1. Bertrand, F., Meyer, N., and Maumy-Bertrand, M. (2015). plsRglm: Partial Least Squares Regression for Generalized Linear Models.
2. Krämer, N., and Sugiyama, N. (2011). The degrees of freedom of partial least squares regression, Journal of the American Statistical Association, 106(494): 697-705.
3. Li, B., Morris, J., and Martin, EB. (2002). Model selection for partial least squares regression. Chemometrics and Intelligent Laboratory Systems, 64: 79-89.
4. Little, R. J. A., and Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd ed.). Wiley, New York.
5. Perry, P. O. (2015). bcv: Cross-Validation for the SVD (Bi-Cross-Validation).
6. Templ, M., Alfons, A., Kowarik, A., and Prantner, B. (2017). VIM: Visualization and Imputation of Missing Values.
7. Van Buuren, S., and Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R, Journal of Statistical Software, 45(3): 1-67.
8. White, I. R., Royston, P., and Wood, A. M. (2011). Multiple imputation using chained equations: Issues and guidance for practice, *Stat. Med.*, vol. 30, no. 4, pp. 377–399.

Figure D.2 – Doctoral School Days.

D.3 The Joint Working Group (JWG) for Cooperation in Higher Education, Research and Innovation

Nous donnons ici le poster présenté dans le cadre des 10èmes Assises France-Indonésie *The Joint Working Group (JWG) for Cooperation in Higher Education, Research and Innovation* qui s'est déroulée à Poitiers, France du 26 au 28 juin 2018.

A Comparison of Determining the Number of Components of a PLS Regression with MCAR mechanism

T.A. Nengsih[1,2], F. Bertrand[2], M. Maumy-Bertrand[2] and N. Meyer [1,3]

[1] ICube, 300 bd Sébastien Brant Illkirch, Université de Strasbourg, Strasbourg, France (E-mail: nengsih@unistra.fr)

[2] IRMA, Université de Strasbourg, 7 rue René-Descartes 67084, Strasbourg, France (E-mail: fbertran@unistra.fr, mmaumy@unistra.fr)

[3] GMRC, Pôle de Santé Publique, CHU de Strasbourg, France, (E-mail: meyer@unistra.fr)

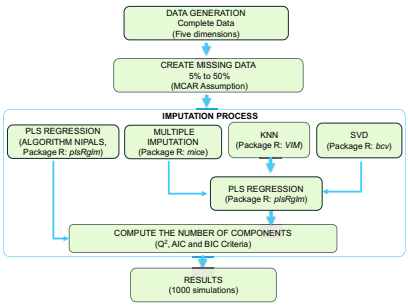
JWG 2018 POITIERS FUTUROSCOPE 26 – 28 June
Cooperation in Higher Education, Research, Innovation and Entrepreneurship
10èmes Assises FRANCE - INDONESIA

INTRODUCTION

Missing data are known to be a concern for the applied research, particularly in health or medical studies. Several methods have been developed for handling incomplete data. The method of imputation is the process of substituting missing data before estimating the relevant model parameters. PLS regression is a multivariate model for which two algorithms (SIMPLS or NIPALS) can be used to provide parameters estimates. PLS regression has been extensively used in the field of health research because of its effectiveness in analysing relationships between the outcome and several components. However, how to handle missing data when using PLS regression is still a matter of debate. The NIPALS algorithm has the interesting property of being able to provide estimates on incomplete data. Selection of the number of components to build a representative model in PLS regression is an important problem. Several approaches have been cited in the literature to choose the number of components to include in a model, such as the Q^2 criterion, the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). The goal of our simulation study was to analyse the impact of the missing data proportion under the missing completely at random (MCAR) assumption on the estimation of the number of components of a PLS regression.

We compared the criteria for selection of the number of components of a PLS regression on incomplete data with NIPALS (NIPALS-PLSR) and PLS regression on imputed data set which used three methods of imputation: multiple imputation by chained equations (MICE), k-nearest neighbor imputation (KNNimpute) and a singular value decomposition imputation (SVDimpute). The criteria which are compared are Q^2-LOO, Q^2-10 fold, AIC, AIC-DoF, BIC and BIC-DoF on different proportions of missing data and under the MCAR assumption. The comparison was performed on different proportions of missing data (ranging from 5% to 50%).

SIMULATION STUDY



1. Data are simulated according to Li *et al.* (2002). The true number of components is set to 4, n (number of observations) and p (number of variables) with each of the following five dimensions sets:
   - n = 100 and p = 20,
   - n = 80 and p = 25,
   - n = 60 and p = 33,
   - n = 40 and p = 50,
   - n = 20 and p = 100.
2. Missing data are created under the MCAR assumption with an increasing proportion of missing data from 5% to 50% by 5% step.
3. Missing data are replaced using MICE, KNNimpute, and SVDimpute.
4. The number of components is computed using LOO (Leave One Out) cross-validation and 10-fold cross-validation computed on the incomplete data according to each of two methods: standard and adaptative (selects the prediction method according to the completeness of the row) (Bertrand *et al.*, 2015). Under MICE, the number of components has been computed by cross-validation as the modal value of the computed number of components across all *m* imputations where *m* is equal to 100 x the proportion of missing data (White *et al.*, 2011).
5. We also computed the maximum number of components that can be extracted for PLS regression, up to 8 components, on incomplete data. We remind that the true number of components is 4.
6. For each combination of the proportion of missing data and matrix dimensions, 1000 replications have been drawn.

RESULTS

Figure 1. The evaluation of Q^2, AIC and BIC with a varying proportion of missing data. The results are expressed as the number of simulations for which the selected components number equals 4 (the true value).

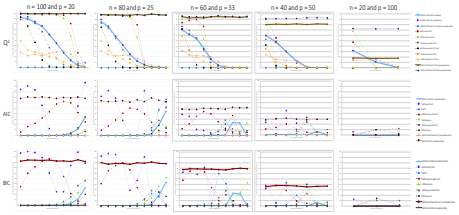


Figure 2. The selected criteria for the number of components of a PLS regression which are close to the true number of components (4) over 1000 simulations.

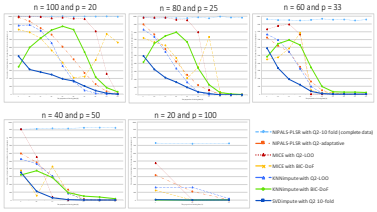
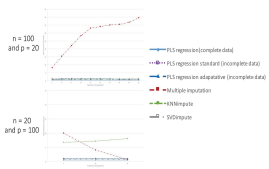


Figure 3. The average of running time (seconds) were calculated over 1000 simulations using 6 * 2793 MHz with 518GB RAM.



On the vertical matrices (n > p):

- The number of selected components by MICE with the Q^2-LOO criterion is much closer to the correct number of components when the proportion of missing data is small.
- The number of selected components using the Q^2-LOO shows a more consistent decreasing pattern of the true number of components for an increasing proportion of missing data and a decreasing sample size.
- The behavior of the different BIC-DoF criterion is not consistent on selecting the true number of components. Their performances increase and then decrease with an increasing proportion of missing data.
- The running time of MICE was about 11 times slower than NIPALS-PLSR for n= 100 and the proportion of missing data= 10%.

On the horizontal matrices (n < p):

- For the smaller sample sizes, it was more difficult to reach convergence of the algorithm, especially on the MICE algorithm.
- A smaller proportion of missing data can make it difficult to estimate the correct number of components in a PLS regression.
- The running time of MICE was about 9 times slower than NIPALS-PLSR for n= 20 and the proportion of missing data= 5%.

CONCLUSIONS

Our simulation study shows that:

- The Q^2-LOO shows the best performance whatever the methods of imputations. The performances increase when the sample size increases and decrease with an increasing proportion of missing data.
- The number of selected components by AIC, AIC-DoF, and BIC is almost twice as large as the true number of components.
- The true number of components of a PLS regression is difficult to determine, especially for small sample size and when the proportion of missing data is larger than 30%.
- The MICE execution took a long time. For example when n= 100 and the proportion of missing data= 10%, the running time of MICE was about 11 times slower than NIPALS-PLSR .

REFERENCES

1. Bertrand, F., Meyer, N., and Maumy-Bertrand, M. (2015). Package 'plsRglm': Partial Least Squares Regression for Generalized Linear Models. URL https://cran.r-project.org/web/packages/ plsRglm/index.html.
2. Krämer, N., and Sugiyama, N. (2011). The degrees of freedom of partial least squares regression, Journal of the American Statistical Association, 106(494): 697-705.
3. Li, B., Morris, J., and Martin, EB. (2002). Model selection for partial least squares regression. Chemometrics and Intelligent Laboratory Systems, 64: 79-89.
4. Little, R. J. A., and Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd ed.). Wiley, New York.
5. Perry, P. O. (2015). Package 'bcv': Cross-Validation for the SVD (Bi-Cross-Validation). URL https://cran.r- project.org/web/packages/bcv/index.html.
6. Templ, M., Alfons, A., Kowarik, A., and Prantner, B. (2017). Package 'VIM': Visualization and Imputation of Missing Values. URL https://github.com/statistikat/VIM.
7. Van Buuren, S., and Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R, Journal of Statistical Software, 45(3): 1-67.
8. White, I. R., Royston, P., and Wood, A. M. (2011). Multiple imputation using chained equations: Issues and guidance for practice, *Stat. Med.*, vol. 30, no. 4, pp. 377–399.

ACKNOWLEDGMENTS

UNIVERSITÉ DE STRASBOURG CNRS IRMA Institut de Recherche Mathématique Avancée Faculté de médecine
ICUBE Laboratoire des sciences de l'ingénieur de l'informatique et de l'imagerie
UNIVERSITAS ISLAM NEGERI SULTHAN THAHA SAIFUDDIN JAMBI
MORA

Figure D.3 – The JWG Indonesia-France Cooperation 2018

D.4 The 23rd International Conference on Computational Statistics

Nous donnons ici le poster présenté dans le cadre du *23rd International Conference on Computational Statistics* qui s'est déroulée à Lasi, Roumanie du 28 au 31 août 2018.

Determining the Number of Components for a PLS Regression on Incomplete Data

T.A. Nengsih[1,2], F. Bertrand[2], M. Maumy-Bertrand[2], and N. Meyer [1,3]

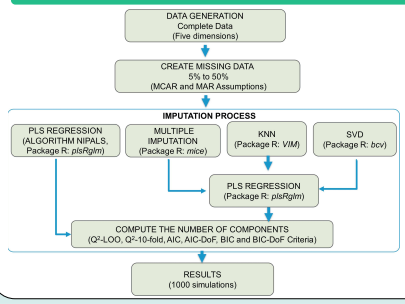
[1] ICube, Université de Strasbourg, 300 bd Sébastien Brant IIIkirch, 67400 Strasbourg, France (E-mail: nengsih@unistra.fr)
[2] IRMA, Université de Strasbourg, 7 rue René-Descartes, 67084 Strasbourg, France (E-mail: fbertran@unistra.fr, mmaumy@unistra.fr)
[3] GMRC, Pôle de Santé Publique, CHU de Strasbourg, France, (E-mail: meyer@unistra.fr)

23rd International Conference on COMPUTATIONAL STATISTICS (COMPSTAT 2018)
28-31 August 2018, Iasi, Romania

INTRODUCTION

Missing data are well-known to be a concern for the applied research. Several methods have been developed for handling incomplete data. The method of imputation is the process of substituting missing data before estimating the relevant model parameters. Partial Least Squares (PLS) regression is a multivariate model estimated either by the SIMPLS or NIPALS algorithm. PLS regression has been extensively used in the applied research because of its effectiveness in analysing relationships between the outcome and several components. However, how to handle missing data when using PLS regression is still a matter of debate. The NIPALS algorithm has the interesting property of being able to provide estimates on incomplete data. Selection of the number of components to build a representative model in PLS regression is an important problem. Several approaches have been cited in the literature to choose the number of components to include in a model, such as the Q^2 criterion, the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC).
The goal of our simulation study was to analyse the impact of the missing data proportion on the estimation of the number of components of a PLS regression under the missing completely at random (MCAR) and the missing at random (MAR) assumptions. We compared the criteria for selection of the number of components of a PLS regression on incomplete data with NIPALS (NIPALS-PLSR) and PLS regression on imputed data set which used three methods of imputation: multiple imputation by chained equations (MICE), k-nearest neighbour imputation (KNNimpute) and a singular value decomposition imputation (SVDimpute). The criteria which are compared are Q^2-LOO, Q^2-10 fold, AIC, AIC-DoF, BIC and BIC-DoF on different proportions of missing data (ranging from 5 to 50%) and under the MCAR and the MAR assumptions.

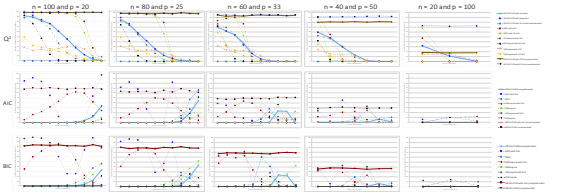
SIMULATION STUDY



1. Data are simulated according to Li et al. (2002) with the true number of components is set to 4. n (number of observations) and p (number of variables) with each of the following five dimensions sets:
   - n = 100 and p = 20,
   - n = 80 and p = 25,
   - n = 60 and p = 33,
   - n = 40 and p = 50,
   - n = 20 and p = 100.
2. Missing data are created under the MCAR and the MAR assumptions with an increasing proportion of missing data from 5% to 50% by 5% step.
3. Missing data are replaced using multiple imputation (MICE), KNNimpute, and SVDimpute.
4. The number of components is computed using LOO (Leave One Out) cross-validation and 10-fold cross-validation computed on the incomplete data according to each of two methods: standard and adaptative (selects the prediction method according to the completeness of the row) (Bertrand et al., 2015). Under multiple imputation, the number of components has been computed by cross-validation as the modal value of the computed number of components across all *m* imputations where *m* is equal to 100 x the proportion of missing data (White et al., 2011).
5. For each combination of the proportion of missing data and matrix dimensions, 1000 replications have been drawn.
6. We also simulated the true number of components which is set to 2 and 6.

MCAR RESULTS

Figure 1. The evaluation of Q^2, AIC and BIC with a varying proportion of missing data. The results are expressed as the number of simulations for which the selected components number equals 4 (the true value).



The results of our study based on seven evaluation criteria with the selected component number equal 2, 4 and 6 (the true value). The number of "+" indicates the performance, from weak (+) to very good one (++++)

n > p

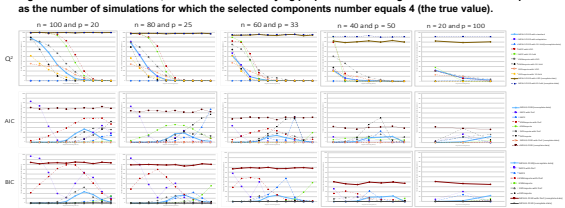
Criteria	Two-Components NIPALS-PLSR	Two-Components MICE	Two-Components KNNimpute	Two-Components SVDimpute	Four-Components NIPALS-PLSR	Four-Components MICE	Four-Components KNNimpute	Four-Components SVDimpute	Six-Components NIPALS-PLSR	Six-Components MICE	Six-Components KNNimpute	Six-Components SVDimpute
Q^2-LOO	++++	+++	++++	+++	++	+++	++	+	+	++	+	+
Q^2-10-fold	+	+++	++++	++++	+	+++	++	+	+	++	+	+
AIC	+	+	+	+	+	+	+	+	+	+	+	+
AIC-DoF	-	++	+	+	-	++	+	+	-	+	+	+
BIC	+	+	+	+	+	+	+	+	+	+	+	+
BIC-DoF	-	+++	+++	++	-	+++	+++	+	-	++	+	+
Execution time	++++	+	++	+++	++++	+	++	+++	++++	+	++	+++

n < p

Criteria	Two-Components NIPALS-PLSR	Two-Components MICE	Two-Components KNNimpute	Two-Components SVDimpute	Four-Components NIPALS-PLSR	Four-Components MICE	Four-Components KNNimpute	Four-Components SVDimpute	Six-Components NIPALS-PLSR	Six-Components MICE	Six-Components KNNimpute	Six-Components SVDimpute
Q^2-LOO	+++	++	++	++	++	++	++	+	+	+	+	+
Q^2-10-fold	+	++	+++	+++	+	++	+	+	+	+	+	+
AIC	+	+	+	+	+	+	+	+	+	+	+	+
AIC-DoF	-	+	+	+	-	+	+	+	-	+	+	+
BIC	+	+	+	+	+	+	+	+	+	+	+	+
BIC-DoF	-	++	+	+	-	+	+	+	-	+	+	+
Execution time	+++	+	++	++++	+++	+	++	++++	+++	+	++	++++

MAR RESULTS

Figure 2. The evaluation of Q^2, AIC and BIC with a varying proportion of missing data. The results are expressed as the number of simulations for which the selected components number equals 4 (the true value).



The results of our study based on seven evaluation criteria with the selected component number equal 2, 4 and 6 (the true value). The number of "+" indicates the performance, from weak (+) to very good one (++++)

n > p

Criteria	Two-Components NIPALS-PLSR	Two-Components MICE	Two-Components KNNimpute	Two-Components SVDimpute	Four-Components NIPALS-PLSR	Four-Components MICE	Four-Components KNNimpute	Four-Components SVDimpute	Six-Components NIPALS-PLSR	Six-Components MICE	Six-Components KNNimpute	Six-Components SVDimpute
Q^2-LOO	+++	+++	++++	++++	++	+++	++	+	+	++	+	+
Q^2-10-fold	+	+++	++++	+++	+	+++	++	++	+	++	+	+
AIC	+	+	+	+	+	+	+	+	+	+	++	+
AIC-DoF	-	++	+	+	-	++	+	+	-	+	+	+
BIC	+	+	+	+	+	+	+	+	+	+	++	+
BIC-DoF	-	+++	+++	+	-	+++	+++	+	-	+	+	+
Execution time	++++	+	+++	++++	++++	+	+++	++++	++++	+	+++	++++

n < p

Criteria	Two-Components NIPALS-PLSR	Two-Components MICE	Two-Components KNNimpute	Two-Components SVDimpute	Four-Components NIPALS-PLSR	Four-Components MICE	Four-Components KNNimpute	Four-Components SVDimpute	Six-Components NIPALS-PLSR	Six-Components MICE	Six-Components KNNimpute	Six-Components SVDimpute
Q^2-LOO	+	++	+++	+++	+	++	++	+	+	+	+	+
Q^2-10-fold	+	++	+++	+++	+	+	+	+	+	+	+	+
AIC	+	+	+	+	+	+	+	+	+	+	+	+
AIC-DoF	-	+	+	+	-	+	+	+	-	+	+	+
BIC	+	+	+	+	+	+	+	+	+	+	+	+
BIC-DoF	-	+	+	+	-	+	+	+	-	+	+	+
Execution time	++++	+	+++	++	++++	+	+++	++	++++	+	+++	++

CONCLUSIONS

Our simulation study shows that:

- Based on four-components, the number of selected components by MICE with Q^2-LOO criterion is much closer to the true number of components when the proportion of missing data is small (< 30%) and n > p, followed by NIPALS-PLSR, KNNimpute and SVDimpute for both the MCAR and the MAR assumptions (Figures 1 and 2).
- According to two-components, the number of selected components by NIPALS-PLSR and KNNimpute with Q^2-LOO criterion shows the best performance that performs closer to the true number of components.
- The number of selected components using the Q^2-LOO shows a more consistent decreasing pattern of the true number of components for an increasing proportion of missing data and a decreasing sample size.
- The number of selected components by AIC, AIC-DoF and BIC are almost twice as large as the true number of components.
- The behaviors of BIC-DoF criterion is not consistent in selecting the true number of components. Their performances increase and then decrease with an increasing proportion of missing data.
- The MICE execution took a long time. For example when n= 100, the proportion of missing data= 10% and under MCAR assumption, the running time of MICE was about 11 times slower than NIPALS-PLSR .
- Whatever the criterion used, the missingness mechanism is also to be considered since it influences the number of selected components.
- The true number of components of a PLS regression is difficult to determine, especially for small sample size and when the proportion of missing data is larger than 30%.

REFERENCES

1. Bertrand, F., Meyer, N., and Maumy-Bertrand, M. (2015). Package 'plsRglm': Partial Least Squares Regression for Generalized Linear Models. URL https://cran.r-project.org/web/packages/ plsRglm/index.html.
2. Krämer, N., and Sugiyama, N. (2011). The degrees of freedom of partial least squares regression, Journal of the American Statistical Association, 106(494): 697-705.
3. Little, R. J. A., and Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd ed.). Wiley, New York.
4. Perry, P. O. (2015). Package 'bcv': Cross-Validation for the SVD (Bi-Cross-Validation). URL https://cran.r- project.org/web/packages/bcv/index.html.
5. Templ, M., Alfons, A., Kowarik, A., and Prantner, B. (2017). Package 'VIM': Visualization and Imputation of Missing Values. URL https://github.com/statistikat/VIM.
6. Van Buuren, S., and Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R, Journal of Statistical Software, 45(3): 1-67.
7. White, I. R., Royston, P., and Wood, A. M. (2011). Multiple imputation using chained equations: Issues and guidance for practice, *Stat. Med.*, vol. 30, no. 4, pp. 377–399.

ACKNOWLEDGMENTS

UNIVERSITÉ DE STRASBOURG · CNRS · IRMA Institut de Recherche Mathématique Avancée · Faculté de médecine · ICUBE Laboratoire des sciences de l'ingénieur de l'informatique et de l'imagerie · UNIVERSITAS ISLAM NEGERI SULTHAN THAHA SAIFUDDIN JAMBI · HCRA

Figure D.4 – The 23rd International Conference on Computational Statistics.

D.5 Statlearn 2019

Nous donnons ici le poster présenté dans le cadre du *19th annual conference of the European Network for Business and Industrial Statistics* qui s'est déroulée à Budapest, Bulgarie du 2 au 4 septembre 2019.

The Performance of Different Algorithms to Determine the Number of Components for a PLS Regression on MCAR and MAR mechanism

T.A. Nengsih[1,2], F. Bertrand[2], M. Maumy-Bertrand[2], and N. Meyer [1,3]

[1] ICube, Université de Strasbourg, 300 bd Sébastien Brant Illkirch, 67400 Strasbourg, France
(E-mail: nengsih@unistra.fr)
[2] IRMA, Université de Strasbourg, 7 rue René-Descartes, 67084 Strasbourg, France
(E-mail: fbertran@unistra.fr, mmaumy@unistra.fr)
[3] GMRC, Pôle de Santé Publique, CHU de Strasbourg, France,
(E-mail: nmeyer@unistra.fr)

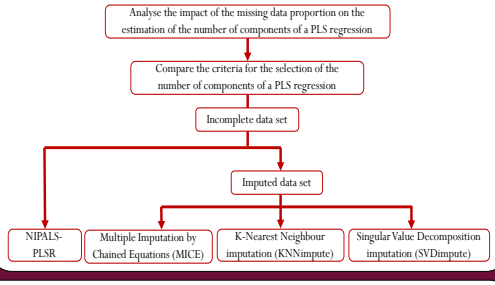
MOTIVATION

Missing data are well-known to be a concern for the applied research. Several methods have been developed for handling incomplete data. The method of imputation is the process of substituting missing data before estimating the relevant model parameters.
Partial Least Squares (PLS) regression is a multivariate model estimated either by the SIMPLS or the NIPALS algorithm. PLS regression has been extensively used in the applied research because of its effectiveness in analysing relationships between the outcome and several components. However, how to handle missing data when using PLS regression is still a matter of debate.
The NIPALS algorithm, without neither imputing the missing data nor deleting the incomplete data rows, has the interesting property of being able to provide estimates on incomplete data. Selection of the number of components to build a representative model in PLS regression remains an important problem.

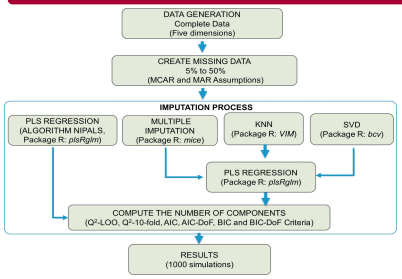
NIPALS Partial Least Squares Regression (NIPALS-PLSR) is employed if

- The explanatory variables are strongly correlated
- The number of the explanatory variables exceeds the size of the sample
- The data base is missing values

GOAL

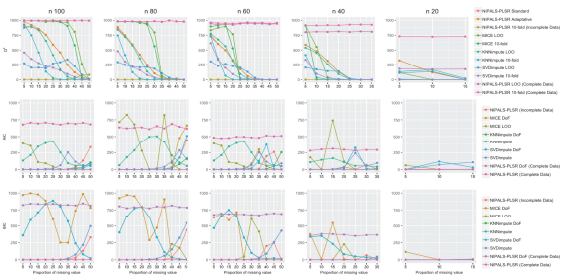


SIMULATION STUDY

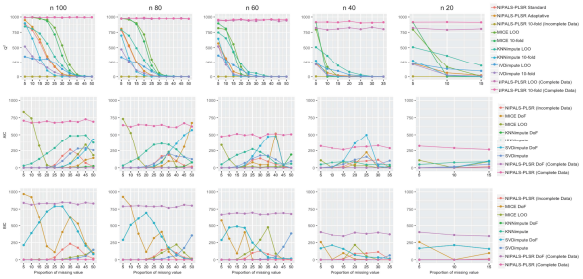


1. Data are simulated according to Li et al. (2002) with the true number of components set to 4. n (number of observations) and p (number of variables) with each of the following five dimensions sets:
   - n = 100 and p = 20,
   - n = 80 and p = 25,
   - n = 60 and p = 33,
   - n = 40 and p = 50,
   - n = 20 and p = 100.
2. Missing data are created under the MCAR and the MAR assumptions with an increasing proportion of missing data from 5% to 50% by 5% step.
3. Missing data are replaced using multiple imputation (MICE), KNNimpute, and SVDimpute.
4. The number of components is computed using LOO (Leave One Out) cross-validation and 10-fold cross-validation computed on the incomplete data according to each of two methods: standard and adaptative (selects the prediction method according to the completeness of the row) (Bertrand et al., 2015). Under multiple imputation, the number of components has been computed by cross-validation as the modal value of the computed number of components across all *m* imputations where *m* is equal to 100 x the proportion of missing data (White et al., 2011).
5. For each combination of the proportion of missing data and matrix dimensions, 1000 replications have been drawn.

MCAR RESULTS



MAR RESULTS



The results under MCAR or MAR mechanism :

- ❖The number of selected components by MICE with Q2-LOO criterion is much closer to the true number of components when the proportion of missing data is small (< 30%) and n > p, followed by NIPALS-PLSR, KNNimpute and SVDimpute for both the MCAR and the MAR assumptions.
- ❖The number of selected components using the Q2-LOO shows a more consistent decreasing pattern of the true number of components for an increasing proportion of missing data and a decreasing sample size.
- ❖The number of selected components by AIC, AIC-DoF and BIC are almost twice as large as the true number of components.
- ❖The behaviors of BIC-DoF to select the true number of components criterion is not consistent. Their performances increase and then decrease with an increasing proportion of missing data.

CONCLUSIONS

Our simulation study shows that:

- The MICE execution took a long time. For example when n= 100, the proportion of missing data= 10% and under MCAR assumption, the running time of MICE was about 11 times slower than NIPALS-PLSR .
- Whatever the criterion used, the missingness mechanism is also to be considered since it influences the number of selected components.
- The true number of components of a PLS regression is difficult to determine, especially for small sample size and when the proportion of missing data is larger than 30%.

REFERENCES

1. Bertrand, F., Meyer, N., and Maumy-Bertrand, M. (2015). Package 'plsRglm': Partial Least Squares Regression for Generalized Linear Models. URL https://cran.r-project.org/web/packages/ plsRglm/index.html.
2. Krämer, N., and Sugiyama, N. (2011). The degrees of freedom of partial least squares regression, Journal of the American Statistical Association, 106(494): 697-705.
3. Little, R. J. A., and Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd ed.). Wiley, New York.
4. Perry, P. O. (2015). Package 'bcv': Cross-Validation for the SVD (Bi-Cross-Validation). URL https://cran.r- project.org/web/packages/bcv/index.html.
5. Templ, M., Alfons, A., Kowarik, A., and Prantner, B. (2017). Package 'VIM': Visualization and Imputation of Missing Values. URL https://github.com/statistikat/VIM.
6. Van Buuren, S., and Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R, Journal of Statistical Software, 45(3): 1-67.
7. White, I. R., Royston, P., and Wood, A. M. (2011). Multiple imputation using chained equations: Issues and guidance for practice, *Stat. Med.*, vol. 30, no. 4, pp. 377–399.

ACKNOWLEDGMENTS

Laboratoire des sciences de l'ingénieur de l'informatique et de l'imagerie

Figure D.5 – Statlearn 2019.

Annexe E

Évaluation du nombre de composantes retenues lors des simulations de la régression *PLS* probabiliste

Dans cette annexe, nous fournissons les résultats des simulations selon les critères et le nombre vrai de composantes en fonction des mécanismes et de la proportion de données manquantes.

Table E.1 – Nombre de composantes retenues pour $t^* = 2$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MCAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	0	98	0	0	0	2	0	3.08
	BIC	0	0	0	98	0	0	0	2	0	3.08
10	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	0	99	1	0	0	0	0	0	2.01
	AIC	0	0	0	0	0	0	0	100	0	7.00
	BIC	0	0	0	0	0	0	0	100	0	7.00
15	Q^2-*LOO*	98	0	2	0	0	0	0	0	0	0.04
	Q^2-10-*Fold*	98	0	1	1	0	0	0	0	0	0.05
	AIC	0	0	0	0	0	98	1	1	0	5.03
	BIC	0	0	0	0	0	98	1	1	0	5.03
20	Q^2-*LOO*	0	0	2	98	0	0	0	0	0	2.98
	Q^2-10-*Fold*	0	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	0	99	1	0	0	0	0	3.01
	BIC	0	0	0	99	1	0	0	0	0	3.01
25	Q^2-*LOO*	0	1	99	0	0	0	0	0	0	1.99
	Q^2-10-*Fold*	0	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	98	0	2	0	0	0	0	2.04
	BIC	0	0	98	0	2	0	0	0	0	2.04
30	Q^2-*LOO*	1	98	1	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	1	98	0	1	0	0	0	0	0	1.01
	AIC	0	1	0	98	0	0	0	0	1	3.03
	BIC	0	1	0	98	0	0	0	0	1	3.03
35	Q^2-*LOO*	0	1	99	0	0	0	0	0	0	1.99
	Q^2-10-*Fold*	0	1	98	0	1	0	0	0	0	2.01
	AIC	0	0	0	98	1	0	1	0	0	3.04
	BIC	0	0	0	98	1	0	1	0	0	3.04
40	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	0	0	99	0	1	0	0	4.02
	BIC	0	0	0	0	99	0	1	0	0	4.02
45	Q^2-*LOO*	0	1	0	99	0	0	0	0	0	2.98
	Q^2-10-*Fold*	0	1	98	1	0	0	0	0	0	2.00
	AIC	0	0	100	0	0	0	0	0	0	2.00
	BIC	0	0	100	0	0	0	0	0	0	2.00
50	Q^2-*LOO*	0	1	99	0	0	0	0	0	0	1.99
	Q^2-10-*Fold*	0	1	99	0	0	0	0	0	0	1.99
	AIC	0	0	1	0	0	98	1	0	0	4.98
	BIC	0	0	1	0	0	98	1	0	0	4.98

Table E.2 – Nombre de composantes retenues pour $t^* = 2$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	1	0	0	0	0	99	0	6.95
	BIC	0	0	1	0	0	0	0	99	0	6.95
10	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	0	2	98	0	0	0	0	0	2.98
	AIC	0	0	0	0	0	1	0	99	0	6.98
	BIC	0	0	0	0	0	1	0	99	0	6.98
15	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	0	0	100	0	0	0	0	0	3.00
	AIC	0	0	0	1	0	98	0	1	0	5.00
	BIC	0	0	0	1	0	98	0	1	0	5.00
20	Q^2-*LOO*	0	0	2	0	0	98	0	0	0	4.94
	Q^2-10-*Fold*	0	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	0	0	0	0	0	99	1	7.01
	BIC	0	0	0	0	0	0	0	99	1	7.01
25	Q^2-*LOO*	0	0	99	1	0	0	0	0	0	2.01
	Q^2-10-*Fold*	0	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	1	98	1	0	0	0	0	3.00
	BIC	0	0	1	98	1	0	0	0	0	3.00
30	Q^2-*LOO*	0	1	99	0	0	0	0	0	0	1.99
	Q^2-10-*Fold*	0	0	2	98	0	0	0	0	0	2.98
	AIC	0	0	0	0	100	0	0	0	0	4.00
	BIC	0	0	0	0	100	0	0	0	0	4.00
35	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	1	98	0	1	0	0	0	0	2.01
	AIC	0	0	0	1	1	0	0	0	98	7.91
	BIC	0	0	0	1	1	0	0	0	98	7.91
40	Q^2-*LOO*	99	0	1	0	0	0	0	0	0	0.02
	Q^2-10-*Fold*	99	1	0	0	0	0	0	0	0	0.01
	AIC	0	0	0	0	100	0	0	0	0	4.00
	BIC	0	0	0	0	100	0	0	0	0	4.00
45	Q^2-*LOO*	1	98	1	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	1	98	0	1	0	0	0	0	0	1.01
	AIC	0	0	1	98	1	0	0	0	0	3.00
	BIC	0	0	1	98	1	0	0	0	0	3.00
50	Q^2-*LOO*	98	2	0	0	0	0	0	0	0	0.02
	Q^2-10-*Fold*	98	2	0	0	0	0	0	0	0	0.02
	AIC	0	0	0	0	1	99	0	0	0	4.99
	BIC	0	0	0	0	1	99	0	0	0	4.99

Table E.3 – Nombre de composantes retenues pour $t^* = 3$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MCAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	39	61	0	0	0	0	0	0	1.61
	Q^2-10-*Fold*	0	59	22	19	0	0	0	0	0	1.60
	AIC	0	0	20	39	40	1	0	0	0	3.22
	BIC	0	0	20	39	40	1	0	0	0	3.22
10	Q^2-*LOO*	0	1	98	1	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	51	49	0	0	0	0	0	0	1.49
	AIC	0	0	0	49	1	0	0	49	1	5.02
	BIC	0	0	0	49	1	0	0	49	1	5.02
15	Q^2-*LOO*	0	98	1	0	0	1	0	0	0	1.05
	Q^2-10-*Fold*	0	1	98	1	0	0	0	0	0	2.00
	AIC	0	0	0	0	97	1	1	0	1	4.07
	BIC	0	0	0	0	97	1	1	0	1	4.07
20	Q^2-*LOO*	0	99	1	0	0	0	0	0	0	1.01
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	1	98	0	1	0	0	0	3.01
	BIC	0	0	1	98	0	1	0	0	0	3.01
25	Q^2-*LOO*	0	2	98	0	0	0	0	0	0	1.98
	Q^2-10-*Fold*	0	2	98	0	0	0	0	0	0	1.98
	AIC	0	0	99	1	0	0	0	0	0	2.01
	BIC	0	0	99	1	0	0	0	0	0	2.01
30	Q^2-*LOO*	0	98	2	0	0	0	0	0	0	1.02
	Q^2-10-*Fold*	0	98	2	0	0	0	0	0	0	1.02
	AIC	0	0	0	0	100	0	0	0	0	4.00
	BIC	0	0	0	0	100	0	0	0	0	4.00
35	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	0	98	0	1	0	0	1	3.07
	BIC	0	0	0	98	0	1	0	0	1	3.07
40	Q^2-*LOO*	0	1	99	0	0	0	0	0	0	1.99
	Q^2-10-*Fold*	1	1	98	0	0	0	0	0	0	1.97
	AIC	0	0	0	0	0	1	98	1	0	6.00
	BIC	0	0	0	0	0	1	98	1	0	6.00
45	Q^2-*LOO*	0	2	98	0	0	0	0	0	0	1.98
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	0	1	99	0	0	0	0	3.99
	BIC	0	0	0	1	99	0	0	0	0	3.99
50	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	0	0	0	99	1	0	0	5.01
	BIC	0	0	0	0	0	99	1	0	0	5.01

Table E.4 – Nombre de composantes retenues pour $t^* = 3$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	51	48	0	1	0	0	0	0	1.51
	Q^2-10-*Fold*	0	51	49	0	0	0	0	0	0	1.49
	AIC	0	0	1	0	1	51	47	0	0	5.43
	BIC	0	0	1	0	1	51	47	0	0	5.43
10	Q^2-*LOO*	0	48	52	0	0	0	0	0	0	1.52
	Q^2-10-*Fold*	0	49	51	0	0	0	0	0	0	1.51
	AIC	0	0	0	0	0	1	49	0	50	6.99
	BIC	0	0	0	0	0	1	49	0	50	6.99
15	Q^2-*LOO*	0	99	1	0	0	0	0	0	0	1.01
	Q^2-10-*Fold*	0	99	1	0	0	0	0	0	0	1.01
	AIC	0	0	0	0	1	0	1	97	1	6.97
	BIC	0	0	0	0	1	0	1	97	1	6.97
20	Q^2-*LOO*	0	73	27	0	0	0	0	0	0	1.27
	Q^2-10-*Fold*	0	74	0	26	0	0	0	0	0	1.52
	AIC	0	0	0	25	0	0	1	74	0	5.99
	BIC	0	0	0	25	0	0	1	74	0	5.99
25	Q^2-*LOO*	0	1	99	0	0	0	0	0	0	1.99
	Q^2-10-*Fold*	0	0	2	98	0	0	0	0	0	2.98
	AIC	0	0	98	0	0	0	0	1	1	2.11
	BIC	0	0	98	0	0	0	0	1	1	2.11
30	Q^2-*LOO*	0	51	0	49	0	0	0	0	0	1.98
	Q^2-10-*Fold*	0	51	49	0	0	0	0	0	0	1.49
	AIC	0	0	98	0	1	1	0	0	0	2.05
	BIC	0	0	98	0	1	1	0	0	0	2.05
35	Q^2-*LOO*	0	2	98	0	0	0	0	0	0	1.98
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	1	0	1	0	97	1	0	5.95
	BIC	0	0	1	0	1	0	97	1	0	5.95
40	Q^2-*LOO*	75	24	1	0	0	0	0	0	0	0.26
	Q^2-10-*Fold*	75	24	1	0	0	0	0	0	0	0.26
	AIC	0	75	2	0	2	0	21	0	0	2.13
	BIC	0	75	2	0	2	0	21	0	0	2.13
45	Q^2-*LOO*	1	1	50	48	0	0	0	0	0	2.45
	Q^2-10-*Fold*	0	99	1	0	0	0	0	0	0	1.01
	AIC	0	0	0	48	1	50	0	0	1	4.06
	BIC	0	0	0	48	1	50	0	0	1	4.06
50	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	98	0	0	1	0	0	1	2.09
	BIC	0	0	98	0	0	1	0	0	1	2.09

Table E.5 – Nombre de composantes retenues pour $t^* = 4$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MCAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	25	75	0	0	0	0	0	0	1.75
	Q^2-10-*Fold*	0	24	27	49	0	0	0	0	0	2.25
	AIC	0	0	0	26	1	24	24	25	0	5.21
	BIC	0	0	0	26	1	24	24	25	0	5.21
10	Q^2-*LOO*	0	33	67	0	0	0	0	0	0	1.67
	Q^2-10-*Fold*	0	33	66	1	0	0	0	0	0	1.68
	AIC	0	0	0	1	0	0	65	0	34	6.65
	BIC	0	0	0	1	0	0	65	0	34	6.65
15	Q^2-*LOO*	0	35	49	16	0	0	0	0	0	1.81
	Q^2-10-*Fold*	0	53	47	0	0	0	0	0	0	1.47
	AIC	0	0	0	1	31	0	34	34	0	5.69
	BIC	0	0	0	1	31	0	34	34	0	5.69
20	Q^2-*LOO*	0	11	75	14	0	0	0	0	0	2.03
	Q^2-10-*Fold*	0	86	14	0	0	0	0	0	0	1.14
	AIC	0	0	1	11	74	2	0	0	12	4.37
	BIC	0	0	1	11	74	2	0	0	12	4.37
25	Q^2-*LOO*	1	97	0	2	0	0	0	0	0	1.03
	Q^2-10-*Fold*	0	97	3	0	0	0	0	0	0	1.03
	AIC	0	0	96	1	1	0	0	2	0	2.13
	BIC	0	0	96	1	1	0	0	2	0	2.13
30	Q^2-*LOO*	0	72	28	0	0	0	0	0	0	1.28
	Q^2-10-*Fold*	0	99	1	0	0	0	0	0	0	1.01
	AIC	0	0	0	1	0	27	72	0	0	5.70
	BIC	0	0	0	1	0	27	72	0	0	5.70
35	Q^2-*LOO*	0	23	77	0	0	0	0	0	0	1.77
	Q^2-10-*Fold*	0	22	78	0	0	0	0	0	0	1.78
	AIC	0	0	0	0	23	0	76	1	0	5.55
	BIC	0	0	0	0	23	0	76	1	0	5.55
40	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	2	0	98	0	0	0	0	0	2.96
	AIC	0	0	98	0	0	0	2	0	0	2.08
	BIC	0	0	98	0	0	0	2	0	0	2.08
45	Q^2-*LOO*	0	98	2	0	0	0	0	0	0	1.02
	Q^2-10-*Fold*	0	98	1	1	0	0	0	0	0	1.03
	AIC	0	0	1	1	0	0	0	0	98	7.89
	BIC	0	0	1	1	0	0	0	0	98	7.89
50	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	2	0	0	98	0	0	0	0	3.94
	AIC	0	0	1	0	0	0	98	0	1	5.98
	BIC	0	0	1	0	0	0	98	0	1	5.98

Table E.6 – Nombre de composantes retenues pour $t^* = 4$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	64	36	0	0	0	0	0	0	1.36
	Q^2-10-*Fold*	0	35	35	30	0	0	0	0	0	1.95
	AIC	0	0	0	5	31	2	60	1	1	5.24
	BIC	0	0	0	5	31	2	60	1	1	5.24
10	Q^2-*LOO*	0	64	20	16	0	0	0	0	0	1.52
	Q^2-10-*Fold*	0	64	18	18	0	0	0	0	0	1.54
	AIC	0	0	0	1	35	1	30	1	32	5.91
	BIC	0	0	0	1	35	1	30	1	32	5.91
15	Q^2-*LOO*	0	9	91	0	0	0	0	0	0	1.91
	Q^2-10-*Fold*	0	15	85	0	0	0	0	0	0	1.85
	AIC	0	0	1	84	7	7	0	1	0	3.24
	BIC	0	0	1	84	7	7	0	1	0	3.24
20	Q^2-*LOO*	0	51	0	49	0	0	0	0	0	1.98
	Q^2-10-*Fold*	0	51	49	0	0	0	0	0	0	1.49
	AIC	0	0	49	1	0	0	2	48	0	4.49
	BIC	0	0	49	1	0	0	2	48	0	4.49
25	Q^2-*LOO*	1	1	97	0	1	0	0	0	0	1.99
	Q^2-10-*Fold*	1	93	6	0	0	0	0	0	0	1.05
	AIC	0	0	0	3	95	0	0	2	0	4.03
	BIC	0	0	0	3	95	0	0	2	0	4.03
30	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	99	1	0	0	0	0	0	0	1.01
	AIC	0	0	0	97	0	1	1	1	0	3.09
	BIC	0	0	0	97	0	1	1	1	0	3.09
35	Q^2-*LOO*	0	98	2	0	0	0	0	0	0	1.02
	Q^2-10-*Fold*	0	99	1	0	0	0	0	0	0	1.01
	AIC	0	0	1	0	0	1	0	98	0	6.93
	BIC	0	0	1	0	0	1	0	98	0	6.93
40	Q^2-*LOO*	0	98	2	0	0	0	0	0	0	1.02
	Q^2-10-*Fold*	1	98	1	0	0	0	0	0	0	1.00
	AIC	0	0	49	0	1	1	49	0	0	4.01
	BIC	0	0	49	0	1	1	49	0	0	4.01
45	Q^2-*LOO*	1	96	3	0	0	0	0	0	0	1.02
	Q^2-10-*Fold*	2	66	32	0	0	0	0	0	0	1.30
	AIC	0	0	2	1	31	0	0	0	66	6.59
	BIC	0	0	2	1	31	0	0	0	66	6.59
50	Q^2-*LOO*	1	99	0	0	0	0	0	0	0	0.99
	Q^2-10-*Fold*	1	99	0	0	0	0	0	0	0	0.99
	AIC	0	0	0	98	1	1	0	0	0	3.03
	BIC	0	0	0	98	1	1	0	0	0	3.03

Table E.7 – Nombre de composantes retenues pour $t^* = 5$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MCAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	3	97	0	0	0	0	0	0	1.97
	Q^2-10-*Fold*	0	1	3	96	0	0	0	0	0	2.95
	AIC	0	0	0	1	3	1	95	0	0	5.90
	BIC	0	0	0	1	3	1	95	0	0	5.90
10	Q^2-*LOO*	0	17	81	1	1	0	0	0	0	1.86
	Q^2-10-*Fold*	0	34	64	2	0	0	0	0	0	1.68
	AIC	0	0	0	16	19	17	46	2	0	4.99
	BIC	0	0	0	16	19	17	46	2	0	4.99
15	Q^2-*LOO*	0	6	94	0	0	0	0	0	0	1.94
	Q^2-10-*Fold*	0	5	95	0	0	0	0	0	0	1.95
	AIC	0	0	96	1	0	0	2	0	1	2.15
	BIC	0	0	96	1	0	0	2	0	1	2.15
20	Q^2-*LOO*	0	52	25	23	0	0	0	0	0	1.71
	Q^2-10-*Fold*	0	7	69	24	0	0	0	0	0	2.17
	AIC	0	0	23	22	8	1	23	23	0	4.48
	BIC	0	0	23	22	8	1	23	23	0	4.48
25	Q^2-*LOO*	0	50	24	1	25	0	0	0	0	2.01
	Q^2-10-*Fold*	0	74	25	1	0	0	0	0	0	1.27
	AIC	0	0	0	27	25	0	24	0	24	5.17
	BIC	0	0	0	27	25	0	24	0	24	5.17
30	Q^2-*LOO*	0	52	48	0	0	0	0	0	0	1.48
	Q^2-10-*Fold*	42	52	5	1	0	0	0	0	0	0.65
	AIC	0	0	2	43	3	43	1	0	8	4.30
	BIC	0	0	2	43	3	43	1	0	8	4.30
35	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	1	99	0	0	0	0	0	0	1.99
	AIC	0	0	0	2	0	0	0	0	98	7.90
	BIC	0	0	0	2	0	0	0	0	98	7.90
40	Q^2-*LOO*	0	98	2	0	0	0	0	0	0	1.02
	Q^2-10-*Fold*	0	98	1	0	1	0	0	0	0	1.04
	AIC	0	0	0	0	97	2	1	0	0	4.04
	BIC	0	0	0	0	97	2	1	0	0	4.04
45	Q^2-*LOO*	0	99	1	0	0	0	0	0	0	1.01
	Q^2-10-*Fold*	0	1	1	0	0	98	0	0	0	4.93
	AIC	0	0	98	0	1	1	0	0	0	2.05
	BIC	0	0	98	0	1	1	0	0	0	2.05
50	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	0	98	1	1	0	0	0	3.03
	BIC	0	0	0	98	1	1	0	0	0	3.03

Table E.8 – Nombre de composantes retenues pour $t^* = 5$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	6	94	0	0	0	0	0	0	1.94
	Q^2-10-*Fold*	0	9	91	0	0	0	0	0	0	1.91
	AIC	0	0	89	2	3	4	1	1	0	2.29
	BIC	0	0	89	2	3	4	1	1	0	2.29
10	Q^2-*LOO*	0	26	74	0	0	0	0	0	0	1.74
	Q^2-10-*Fold*	0	73	25	1	1	0	0	0	0	1.30
	AIC	0	0	0	47	49	0	2	2	0	3.63
	BIC	0	0	0	47	49	0	2	2	0	3.63
15	Q^2-*LOO*	0	7	93	0	0	0	0	0	0	1.93
	Q^2-10-*Fold*	0	9	90	1	0	0	0	0	0	1.92
	AIC	0	0	4	0	2	85	8	1	0	4.96
	BIC	0	0	4	0	2	85	8	1	0	4.96
20	Q^2-*LOO*	0	51	49	0	0	0	0	0	0	1.49
	Q^2-10-*Fold*	0	61	39	0	0	0	0	0	0	1.39
	AIC	0	0	12	12	3	0	24	36	13	5.72
	BIC	0	0	12	12	3	0	24	36	13	5.72
25	Q^2-*LOO*	0	99	1	0	0	0	0	0	0	1.01
	Q^2-10-*Fold*	0	99	1	0	0	0	0	0	0	1.01
	AIC	0	0	49	1	1	0	49	0	0	3.99
	BIC	0	0	49	1	1	0	49	0	0	3.99
30	Q^2-*LOO*	0	51	49	0	0	0	0	0	0	1.49
	Q^2-10-*Fold*	0	52	48	0	0	0	0	0	0	1.48
	AIC	0	1	0	24	1	48	24	1	1	4.76
	BIC	0	1	0	24	1	48	24	1	1	4.76
35	Q^2-*LOO*	0	36	64	0	0	0	0	0	0	1.64
	Q^2-10-*Fold*	0	37	63	0	0	0	0	0	0	1.63
	AIC	0	0	0	2	66	1	1	30	0	4.91
	BIC	0	0	0	2	66	1	1	30	0	4.91
40	Q^2-*LOO*	32	35	33	0	0	0	0	0	0	1.01
	Q^2-10-*Fold*	32	34	34	0	0	0	0	0	0	1.02
	AIC	0	32	1	1	0	33	1	1	31	4.63
	BIC	0	32	1	1	0	33	1	1	31	4.63
45	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	1	3	1	0	95	0	0	0	4.85
	BIC	0	1	3	1	0	95	0	0	0	4.85
50	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	0	1	97	2	0	0	0	4.01
	BIC	0	0	0	1	97	2	0	0	0	4.01

Table E.9 – Nombre de composantes retenues pour $t^* = 6$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MCAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	95	5	0	0	0	0	0	0	1.05
	Q^2-10-*Fold*	0	3	96	1	0	0	0	0	0	1.98
	AIC	0	0	0	1	3	93	1	2	0	5.00
	BIC	0	0	0	1	3	93	1	2	0	5.00
10	Q^2-*LOO*	0	84	15	0	1	0	0	0	0	1.18
	Q^2-10-*Fold*	0	81	17	2	0	0	0	0	0	1.21
	AIC	0	0	2	8	1	79	4	5	1	4.94
	BIC	0	0	2	8	1	79	4	5	1	4.94
15	Q^2-*LOO*	0	41	58	1	0	0	0	0	0	1.60
	Q^2-10-*Fold*	0	41	59	0	0	0	0	0	0	1.59
	AIC	0	0	0	19	3	20	2	18	38	6.11
	BIC	0	0	0	19	3	20	2	18	38	6.11
20	Q^2-*LOO*	0	82	18	0	0	0	0	0	0	1.18
	Q^2-10-*Fold*	0	95	4	1	0	0	0	0	0	1.06
	AIC	0	0	0	74	15	9	1	1	0	3.40
	BIC	0	0	0	74	15	9	1	1	0	3.40
25	Q^2-*LOO*	1	51	48	0	0	0	0	0	0	1.47
	Q^2-10-*Fold*	0	6	94	0	0	0	0	0	0	1.94
	AIC	0	0	0	46	2	48	2	1	1	4.13
	BIC	0	0	0	46	2	48	2	1	1	4.13
30	Q^2-*LOO*	0	95	2	3	0	0	0	0	0	1.08
	Q^2-10-*Fold*	1	92	7	0	0	0	0	0	0	1.06
	AIC	0	0	3	45	46	0	3	3	0	3.64
	BIC	0	0	3	45	46	0	3	3	0	3.64
35	Q^2-*LOO*	0	76	24	0	0	0	0	0	0	1.24
	Q^2-10-*Fold*	0	77	22	1	0	0	0	0	0	1.24
	AIC	0	0	22	2	74	2	0	0	0	3.56
	BIC	0	0	22	2	74	2	0	0	0	3.56
40	Q^2-*LOO*	0	96	4	0	0	0	0	0	0	1.04
	Q^2-10-*Fold*	0	0	5	0	0	0	95	0	0	5.80
	AIC	0	0	0	1	97	1	1	0	0	4.02
	BIC	0	0	0	1	97	1	1	0	0	4.02
45	Q^2-*LOO*	1	3	0	96	0	0	0	0	0	2.91
	Q^2-10-*Fold*	0	2	98	0	0	0	0	0	0	1.98
	AIC	0	0	0	0	98	1	1	0	0	4.03
	BIC	0	0	0	0	98	1	1	0	0	4.03
50	Q^2-*LOO*	96	3	1	0	0	0	0	0	0	0.05
	Q^2-10-*Fold*	96	3	1	0	0	0	0	0	0	0.05
	AIC	0	0	0	97	1	0	0	0	2	3.11
	BIC	0	0	0	97	1	0	0	0	2	3.11

Table E.10 – Nombre de composantes retenues pour $t^* = 6$ et la dimension $n = 100$ et $p = 20$ selon la proportion de données manquantes (d) et les critères sur 100 simulations, sous l'hypothèse *MAR*.

d	Critères	Nombre de composantes retenues									Moyenne du nombre de composantes
		0	1	2	3	4	5	6	7	8	
5	Q^2-*LOO*	0	99	0	1	0	0	0	0	0	1.02
	Q^2-10-*Fold*	0	98	2	0	0	0	0	0	0	1.02
	AIC	0	0	25	25	0	0	24	26	0	4.51
	BIC	0	0	25	25	0	0	24	26	0	4.51
10	Q^2-*LOO*	0	0	100	0	0	0	0	0	0	2.00
	Q^2-10-*Fold*	0	0	100	0	0	0	0	0	0	2.00
	AIC	0	0	0	1	99	0	0	0	0	3.99
	BIC	0	0	0	1	99	0	0	0	0	3.99
15	Q^2-*LOO*	0	64	35	1	0	0	0	0	0	1.37
	Q^2-10-*Fold*	0	66	34	0	0	0	0	0	0	1.34
	AIC	0	0	32	33	1	33	0	1	0	3.39
	BIC	0	0	32	33	1	33	0	1	0	3.39
20	Q^2-*LOO*	0	52	48	0	0	0	0	0	0	1.48
	Q^2-10-*Fold*	0	51	49	0	0	0	0	0	0	1.49
	AIC	0	0	0	51	0	48	1	0	0	3.99
	BIC	0	0	0	51	0	48	1	0	0	3.99
25	Q^2-*LOO*	33	34	33	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	34	66	0	0	0	0	0	0	1.66
	AIC	0	0	65	0	0	2	0	33	0	3.71
	BIC	0	0	65	0	0	2	0	33	0	3.71
30	Q^2-*LOO*	0	99	1	0	0	0	0	0	0	1.01
	Q^2-10-*Fold*	0	2	98	0	0	0	0	0	0	1.98
	AIC	0	0	1	97	0	2	0	0	0	3.03
	BIC	0	0	1	97	0	2	0	0	0	3.03
35	Q^2-*LOO*	0	66	34	0	0	0	0	0	0	1.34
	Q^2-10-*Fold*	0	34	66	0	0	0	0	0	0	1.66
	AIC	0	0	0	34	0	0	33	1	32	5.63
	BIC	0	0	0	34	0	0	33	1	32	5.63
40	Q^2-*LOO*	49	50	1	0	0	0	0	0	0	0.52
	Q^2-10-*Fold*	49	1	0	0	0	0	50	0	0	3.01
	AIC	0	0	0	0	0	0	0	0	100	8.00
	BIC	0	0	0	0	0	0	0	0	100	8.00
45	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	0	0	0	2	98	0	0	5.98
	BIC	0	0	0	0	0	2	98	0	0	5.98
50	Q^2-*LOO*	0	100	0	0	0	0	0	0	0	1.00
	Q^2-10-*Fold*	0	100	0	0	0	0	0	0	0	1.00
	AIC	0	0	0	0	0	50	0	50	0	6.00
	BIC	0	0	0	0	0	50	0	50	0	6.00

Bibliographie

Acazencott, C.-A. (2020). Mettez en place un cadre de validation croisée. https://openclassrooms.com/fr/courses/4297211-evaluez-et-ameliorez-les-performances-dun_-modele-de-machine-learning/4308241-mettez-en-place-un-cadre-de-_validation-croisee, dernier accès le 29-01-2020.

Akaike, H. (1969). Fitting autoregressive models for prediction. *Ann. Ins. Stat. Math*, 21 :243–247.

Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control,*, 19(6) :716–723.

Andersson, M. (2009). A comparison of nine PLS1 algorithms. *Journal of Chemometrics*, 23 :518–529.

Arteaga, F. and Ferrer, A. (2002). Dealing with missing data in MSPC : Several methods, different interpretations, some examples. *Journal of Chemometrics*, 16(8-10) :408–418.

Azur, M. J., Stuart, E. A., Frangakis, C., and Leaf, P. J. (2011). Multiple Imputation by Chained Equations : What is it and how does it work ? *Int J methods Psychiatr Res*, 20(1) :40–49.

Bastien, P., Bertrand, F. F., Meyer, N., and Maumy-Bertrand, M. (2015). Deviance residuals-based sparse PLS and sparse kernel PLS regression for censored data. *Bioinformatics*, 31(3) :397–404.

Bastien, P. and Tenenhaus, M. (2003). PLS Regression and Multiple Imputation. *Proceedings of the PLS'03 International Symposium, Vilares M et al. editors CISIA Paris*, 497-498.

Batista, G. E. A. P. A. and Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6) :519–533.

Bertrand, F., Meyer, N., and Maumy-Bertrand, M. (2014). plsRglm : Partial Least Squares Regression for Generalized Linear Models, book of abstracts, User2014 !, Los Angeles. R package version 1.2.5. https://cran.r-project.org/web/packages/plsRglm/index.html, dernier accès le 29-01-2020.

Besse (2020). Imputation de données manquantes. https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-idm.pdf, dernier accès le 31-01-2020.

Bodner, T. E. (2008). What improves with increased missing data imputations ? *Structural Equation Modeling*, 15 :651–675.

Bouhaddani, S., Uh, H.-W., Hayward, C., and Houwing-Duistermaat, J. (2018). Probabilistic partial least squares model : Identifiability, estimation and application. *Journal of Multivariate Analysis*, 167 :331–346.

Bouhaddani, S. E. (2017). Probabilistic Two-Way Orthogonal Partial Least Squares. https://github.com/selbouhaddani/PO2PLS, dernier accès le 29-01-2020.

Bouhlila, D. S. and Sellaouti, F. (2013). Multiple imputation using chained equations for missing data in TIMSS : a case study. *Large-scale Assessments in Education*, 1(4) :1–4.

Breiman, L. *et al.* (1984a). *Classification and Regression Trees*. Chapman & Hall, New York. new edition of Breiman *et al.* (1984b) ?

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984b). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA. new edition Breiman *et al.* (1984a) ?

Burnham, A. J., Macgregor, J. F., and Viveros, R. (1999). Latent variable multivariate regression modeling. *Chemometrics and Intelligent Laboratory Systems*, 48 :167–180.

Burnham, A. J., Viveros, R., and Macgregor, J. F. (1996). Frameworks for latent variable multivariate regression. *Journal of Chemometrics*, 10 :31–45.

Byrd, A. (2008). *Penalized principal component regression*. Master's thesis, University of Georgia.

De Jong, S. (1993a). PLS fits closer than PCR. *Journal of Chemometrics*, 7(6) :551–557.

De Jong, S. (1993b). SIMPLS : an alternative approach squares regression to partial least. *Chemometrics and Intelligent Laboratory Systems*, 18 :251–263.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1997). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1) :1– 38.

Dixon, J. K. (1979). Pattern Recognition with Partly Missing Data. *IEEE Trans.Syst.Man Cybern*, 10 :617–621.

Eastment, H. T. and Krzanowski, W. J. (1982). Cross-validatory choice of the number of components from a principal component analysis. *Technometrics*, 24(1) :73–77.

Efron, B. (2004). The estimation of prediction error. *Journal of the American Statistical Association*, 99(467) :619–632.

Eriksson, I., Johansson, E., Kettaneh-Wold, N., and Wold, S. (2002). Multi- and megavariate data analysis, principles and applications. *Journal of Chemometrics*, 16 :261–262.

Folch-Fortuny, A., Arteaga, F., and Ferrer, A. (2016). Missing Data Imputation Toolbox for MATLAB. *Chemometrics and Intelligent Laboratory Systems*, 154 :93–100.

Fornecker, L. M., Muller, L., Bertrand, F., Paul, N., Pichot, A., Herbrecht, R., Chenard, M. P., Mauvieux, L., Vallat, L., Bahram, S., Cianférani, S., Carapito, R., and Carapito, C. (2019). Multi-omics dataset to decipher the complexity of drug resistance in diffuse large B-cell lymphoma. *Scientific Reports*, 9(1) :1–9.

Frank, I. E. and Friedman, J. H. (1993). A Statistical view of some chemometrics regression tools. *Technometrics*, 35(2) :109–135.

Goicoechea, H. C. and Olivieri, A. C. (1999a). Determination of bromhexine in cough-cold syrups by absorption spectrophotometry and multivariate calibration using partial least-squares and hybrid linear analyses. Application of a novel method of wavelength selection. *Talanta*, 49(4) :793–800.

Goicoechea, H. C. and Olivieri, A. C. (1999b). Enhanced synchronous spectrofluorometric determination of tetracycline in blood serum by chemometric analysis. Comparison of partial least-squares and hybrid linear analysis calibrations. *Analytical Chemistry*, 71(19) :4361–4368.

Goicoechea, H. C. and Olivieri, A. C. (2003). A new family of genetic algorithms for wavelength interval selection in multivariate analytical spectroscopy. *Journal of Chemometrics*, 17(6) :338–345.

Gourvénec, S., Fernández Pierna, J. A., Massart, D. L., and Rutledge, D. N. (2003). An evaluation of the PoLiSh smoothed regression and the Monte Carlo Cross-Validation for the determination of the complexity of a PLS model. *Chemometrics and Intelligent Laboratory Systems*, 68(1-2) :41–51.

Gower, J. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, 27 :857–871.

Graham, J. W., Olchowski, A. E., and Gilreath, T. D. (2007). How many imputations are really needed ? Some practical clarifications of multiple imputation theory. *Prevention Science*, 8 :206–213.

Grung, B. and Manne, R. (1998). Missing values in principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 42 :125–139.

H. Kutner, M., J. Nachtsheim, C., and Neter, J. (2004). *Applied Linear Regression Models (4th ed.)*. McGraw-Hill Irwin.

Hastie, T., Tibsharani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*, volume 1. Springer New York, 233 Spring Street, New York, NY, 10013, USA, second edition.

Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., and Botstein, D. (1999). Imputing missing data for gene expression arrays. *Technical Report, Division of Biostatistics, Stanford University*, pages 1–9.

Horton, N. J. and Lipsitz, S. R. (2001). Multiple imputation in practice : Comparison of software packages for regression models with missing variables. *The American Statistician*, 55 :244–254.

Höskuldsson, A. (1988). PLS regression. *Journal of Chemometrics*, 2 :211–228.

Hotelling, H. (1957). The relations of the newer multivariate statistical methods to factor analysis. *British Journal of Statistical Psychology*, 10(2) :69–79.

J. Sheather, S. (2009). *A modern approach to regression with R*. New York, NY : Springer.

Jolliffe, I. T. (1982). A Note on the Use of Principal Components in Regression. *Applied Statistics*, 31(3).

Kowarik, A. and Templ, M. (2016). Imputation with the R Package VIM. *Journal of Statistical Software*, 74(7).

Krämer, N. and Braun, M. L. (2019). Degrees of Freedom and Statistical Inference for Partial Least Squares Regression. https://cran.r-project.org/web/packages/plsdof/index.html, dernier accès le 29-01-2020.

Krämer, N. and Sugiyama, M. (2011). The Degrees of Freedom of Partial Least Squares Regression. *Journal of the American Statistical Association*, 106 :697–705.

Kvalheim, O. (1992). The latent variable. *Chemometrics and Intelligent Laboratory Systems*, 14 :1–3.

Lazraq, A., Cléroux, R., and Gauchi, J.-P. (2003). Selecting both latent and explanatory variables in the PLS1 regression model. *Chemometrics and Intelligent Laboratory Systems*, 66 :117–126.

Lee, H., Park, Y. M., and Lee, S. (2015). Principal Component Regression by Principal Component Selection. *Communications for Statistical Applications and Methods*, 22 :173–180.

Leisch, F. and Dimitriadou, E. (2010). Machine Learning Benchmark Problems : Package 'mlbench'. https://cran.r-project.org/web/packages/mlbench/index.html, dernier accès le 29-01-2020.

Li, B., Morris, J., and Martin, E. B. (2002). Model selection for partial least squares regression. *Chemometrics Intell. Lab. Syst.*, 64 :79–89.

Li, S., Gao, J., Nyagilo, J. O., and Dave, D. P. (2011). Probabilistic Partial Least Square Regression : A Robust Model for Quantitative Analysis of Raman Spectroscopy Data. *IEEE International Conference on Bioinformatics and Biomedicine*.

Li, S., Nyagilo, J., Dave, D., Wang, W., Zhang, B., and Gao, J. (2015). Probabilistic partial least squares regression for quantitative analysis of Raman spectra. *Int. J. Data Min. Bioinform*, 11 :223–243.

Liew, A. W.-C., Law, N.-F., and Yan, H. (2011). Missing value imputation for gene expression data : Computational techniques to recover missing data from available information. *Briefings in Bioinformatics*, 12(5) :498–513.

Liquet, B., De Micheaux, P. L., Hejblum, B. P., and Thiébaut, R. (2016). Group and sparse group partial least square approaches applied in genomics context. *Bioinformatics*, 32(1) :35–42.

Little, R. J. and Rubin, D. B. (1987). *Statistical Analysis with Missing Data*. Wiley, New York, Wiley Series in Probability and Statistics - Applied Probability and Statistics Series.

Little, R. J. and Rubin, D. B. (2002). *Statistical analysis with missing data*. A John Wiley & Sons, Inc, New York, 2nd edition.

Mandel J, S. P. (2015). A Comparison of Six Methods for Missing Data Imputation. *Journal of Biometrics & Biostatistics*, 06(01) :1–6.

Merola, G. and Abraham, B. (2003). Dimension reduction methods used in industry. In *Handbook of Statistics*, volume 22, pages 995–1039. Elsevier.

Meyer, N. (2007). *Méthodes statistiques d'analyse des données de allélotypage en présence de homozygotes*. PhD thesis, Université Louis Pasteur Strasbourg I.

Meyer, N., Maumy-Bertrand, M., and Bertrand, F. (2010). Comparaison de variantes de régressions logistiques PLS et de régression PLS sur variables qualitatives : application aux données d'allélotypage. *Journal de la Société Française de Statistique*, 151(2) :1–18.

Naes, T. and Martens, H. (1985). Comparison of prediction methods for multicollinear data. *Commun. Stat., Simul*, 14 :545–576.

Nelson, P. R., Taylor, P. A., and MacGregor, J. F. (1996). Missing data methods in PCA and PLS : Score calculations with incomplete observations. *Chemometrics and Intelligent Laboratory Systems*, 35(1) :45–65.

Nguyen, D. V. and Rocke, D. M. (2004). On partial least squares dimension reduction for microarray-based classification : A simulation study. *Computational Statistics and Data Analysis*, 46 :407–425.

Oleszko, A., Hartwich, J., Wójtowicz, A., Gąsior-Głogowska, M., Huras, H., and Komorowska, M. (2017). Comparison of FTIR-ATR and Raman spectroscopy in determination of VLDL triglycerides in blood serum with PLS regression. *Spectrochimica Acta - Part A : Molecular and Biomolecular Spectroscopy*, 183 :239–246.

Pedersen, A. B., Mikkelsen, E. M., Cronin-Fenton, D., Kristensen, N. R., Pham, T. M., Pedersen, L., and Petersen, I. (2017). Missing data and multiple imputation in clinical epidemiological research. *Clinical Epidemiology*, 9 :157–166.

Pérez-Enciso, M. and Tenenhaus, M. (2003). Prediction of clinical outcome with microarray data : a partial least squares discriminant analysis (PLS-DA) approach. *Hum Genet*, 112 :581–592.

Perry, P. O. (2015). Package 'bcv' : Cross-Validation for the SVD (Bi-Cross-Validation). https://cran.r-project.org/web/packages/bcv/index.html, dernier accès le 29-01-2020.

Rännar, S., Geladi, P., Lindgren, F., and Wold, S. (1995). A PLS Kernel algorithm for data sets with many variables and few objects. 2. Cross-validataion, missing data and examples. *Journal of Chemometrics*, 9 :459–470.

Rosipal, R. and Krämer, N. (2006). Overview and recent advances in partial least squares. In *Subspace, latent structure and feature selection*, pages 34–51. Springer, Springer-Verlag Berlin, Heidelberg, Deutschland.

Royston, P. (2004). Multiple imputation of missing values. *The Stata Journal*, 4(3) :227–241.

Rubin, D. B. (1976). Inference and Missing Data. *Biometrika*, 63(3) :581–592.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Son, New York, New York.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of American Statistical Association*, 91(434) :473–489.

Sawatsky, M. L., Clyde, M., and Meek, F. (2015). Partial Least Squares regression in the social sciences. *The Quantitative Method for Psychology*, 11(2) :52–62.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. A CRC Press Company.

Schafer, J. L. and Graham, J. W. (2002). Missing data : Our view of the state of the art. *Psychological Methods*, 7(2) :147–177.

Schmitt, P., Mandel, J., and Guedj, M. (2015). A Comparison of Six Methods for Missing Data Imputation. *Journal of Biometrics & Biostatistics*, 06(01) :1–6.

Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2) :461–464.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2) :461 – 464.

Serneels, S. and Verdonck, T. (2008). Principal component regression for data containing outliers and missing elements. *Computational Statistics and Data Analysis*, 52 :1712–1727.

Shao, J. (1993). Linear Model Selection by Cross-Validation. *Journal of American Statistical Association*, 88(422) :486–494.

Srinivasan, B. V., Schwartz, W. R., Duraiswami, R., and Davis, L. (2010). Partial least squares on graphical processor for efficient pattern recognition.

Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society*, 36(2) :111–147.

Stone, M. and Brooks, R. (1990). Continuum Regression : Cross-Validated Sequentially Constructed Prediction Embracing Ordinary Least Squares, Partial Least Squares and Principal Components Regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52 :237–269.

Templ, M., Alfons, A., Kowarik, A., and Prantner, B. (2017). Package 'VIM' : Visualization and Imputation of Missing Values. https://cran.r-project.org/web/packages/VIM/VIM.pdf, dernier accès le 29-01-2020.

Tenenhaus, M. (1998). *La Régression PLS : Théorie et Pratique*. Editions Technip, 27, Rue Ginoux 75737, Paris Cedex 1.

The R Core Team (2014). R : A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.r-project.org/, dernier accès le 29-01-2020.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *J. R. Statist. Soc. f*, 61(Part 3) :611 – 622.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics (Oxford, England)*, 17(6) :520–525.

Ullah, M. I. and Aslam, M. (2018). Package 'mctest'. https://cran.r-project.org/web/packages/mctest/index.html, dernier accès le 29-01-2020.

Vallat, L., Kemper, C. A., Jung, N., Maumy-Bertrand, M., Bertrand, F., Meyer, N., Pocheville, A., Fisher, J. W., Gribben, J. G., and Bahram, S. (2013). Reverse-engineering the genetic circuitry of a cancer cell with predicted intervention in chronic lymphocytic leukemia. *Proceedings of the National Academy of Sciences of the United States of America*, 110(2) :459–464.

Van Buuren, S. (2007). Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical Methods in Medical Research*, 16 :219–242.

Van Buuren, S. (2012). *Flexible Imputation of Missing Data*. Chapman & Hall/CRC, Boca Raton.

Van Buuren, S. (2018). Package 'mice'. https://cran.r-project.org/web/packages/mice/mice.pdf, dernier accès le 29-01-2020.

Van Buuren, S. and Groothuis-Oudshoorn, K. (2000). Multivariate Imputation by Chained Equations : MICE V1.0 User's ManualR. *PG/VGZ/00.038. Leiden : TNO Prevention ; Health.*

Van Buuren, S. and Groothuis-Oudshoorn, K. (2011). mice : Multivariate Imputation by Chained Equation in R. *Journal of Statistical Software*, 45(3).

Wakeling, I. N. and Morris, J. J. (1993). A test of significance for partial least squares regression. *Journal of Chemometrics*, 7(1993) :291–304.

White, I. R., Royston, P., and Wood, A. M. (2011). Multiple imputation using chained equations : Issues and guidance for practice. *Statistics in Medicine*, 30(4) :377–399.

Wiklund, S., Nilsson, D., Eriksson, L., Sjöström, M., Wold, S., and Faber, K. (2007). A randomization test for PLS component selection. *Journal of Chemometrics*, 21 :427–439.

Wold, H. (1966). Estimation of principal components and related models by iterative least squares. *Multivariate analysis*, pages 391–420.

Wold, H. (1973). Nonlinear iterative partial least squares (NIPALS) modeling : some current developments. In P. R. Krishnaiah, editor. *Multivariate Analysis II, Proceedings of an international symposium on multivariate analysis held at Wright State University, Dayton, Ohio, June 19-24, 1972*, page 383–407.

Wold, S. (2001). Personal memories of the early PLS development. *Chemometrics and Intelligent Laboratory Systems*, 58(2) :83–84.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1) :37–52.

Wold, S., Kettaneh-Wold, N., and Skagerberg, B. (1989). Nonlinear PLS modeling. *Chemometrics and Intelligent Laboratory Systems*, 7(1) :5365.

Wold, S., Martens, H., and Wold, H. (1983). The multivariate calibration problem in chemistry solved by the PLS method. In Kågström, B. and Ruhe, A., editors, *Matrix Pencils : Proceedings of a Conference Held at Pite Havsbad, Sweden, March 22–24, 1982*, pages 286–293. Springer Berlin Heidelberg, Berlin, Heidelberg.

Wold, S., Ruhe, A., Wold, H., and Dunn III, W. J. (1984). The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM J. Sci. and Stat. Comput.*, 5(3) :735–743.

Wold, S., Sjöström, M., and Eriksson, L. (2001). PLS-regression : A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58 :109–130.

Yang, T. C., Aucott, L. S., Duthie, G. G., and Macdonald, H. M. (2017). An application of partial least squares for identifying dietary patterns in bone health. *Archives of osteoporosis*, 12(1) :63.

Zheng, J., Song, Z., and Ge, Z. (2016). Probabilistic learning of partial least squares regression model : Theory and industrial applications. *Chemom. Intell. Lab. Syst*, 158 :80–90.

Université de Strasbourg

Titin Agustin Nengsih
Robustesse et dimensions des modèles de régression PLS en cas de données incomplètes

École Doctorale des Sciences de la Vie et de la Santé STRASBOURG

Résumé

Dans la recherche et dans le développement, les données manquantes sont un réel problème pour le praticien. Plusieurs approches statistiques ont été développées pour traiter des données manquantes. Les techniques d'imputation consistent à remplacer les données manquantes par une valeur générée au cours d'un processus d'imputation. La régression *PLS* est un modèle multivarié pour lequel deux algorithmes (*SIMPLS* ou *NIPALS*) existent et qui a été largement utilisée en raison de son efficacité dans l'analyse des relations entre plusieurs composantes. L'algorithme *NIPALS* a l'avantage de pouvoir estimer les composantes même lorsque les données sont incomplètes, dans la mesure où chaque composante est estimée à partir des seules données complètes, de manière itérative sur chaque dimension du jeu de données et ceci, sans devoir recourir à l'imputation des éventuelles donnés manquantes. Bien qu'il soit désormais considéré comme une méthode de référence dans le traitement des données incomplètes, les performances de l'algorithme *NIPALS* sont mal connues dans ce cas des données incomplètes. La détermination du nombre de composantes construites lors de la régression *PLS* ne tient pas compte ni du type de manquant ni de la proportion de données manquantes dans le jeu de données. Pourtant il s'agit d'un point essentiel pour établir des modèles de régression fiables ainsi que pour sélectionner correctement des prédicteurs. Dans la détermination du nombre de composantes, plusieurs critères ont été étudiés. Nous avons comparé les performances des critères sur un jeu de données incomplet et sur un jeu de données imputé en utilisant trois méthodes d'imputation : *MICE*, l'imputation *KNN* et l'imputation *SVD*. Nous avons testé plusieurs critères sous différentes hypothèses de type et de proportion de données manquantes et sur des jeux de données de différentes dimensions.

English summary

Missing data are known to be a concern for the applied researcher. Several methods have been developed for handling incomplete data. Method of Imputation is the process of substituting missing data before estimating the relevant model parameters. Furthermore, PLS regression is a multivariate model for which two algorithms (SIMPLS or NIPALS) can be used to provide its parameters estimates. This model has been extensively used in research because of its effectiveness in analyzing relationships between several components. The NIPALS algorithm has the interesting property of being able to provide estimates on incomplete data. However, the NIPALS-PLS algorithm performances are not known when applied to incomplete data. Selection of the number of components to build a representative model in PLS regression is an important problem. Fitting the number of components of a PLS regression on incomplete data set leads to the problem of model validation, which is generally done using one of several criteria with simulations. We compared the criteria for selection of the number of components of a PLS regression according to PLS regression with NIPALS algorithm on incomplete data and PLS regression on imputed data set, applying three methods of imputation: MICE, KNN imputation and SVD imputation. The comparison was performed under different assumptions on proportions of missing data and missingness mechanism, for different dataset dimensions.