Traitement des données manquantes

Projet M1 - S2

FIGURE 1 – Missing data representation

Encadrant : DUCHARME Gilles

Luana TIMOFTE - Anas EL BENNA

Faculté de Sciences Montpellier
Master 1
Biostatistiques
9 Janvier 2019

Table des matières

1 Présentation

1.1 Introduction

Dans le monde des statistiques, nous rencontrons souvent des données (ou valeurs) manquantes (appelées **NA** [1]), lors d'observation des variables et il est très rare d'avoir un jeu de données complet, sans aucune valeur manquante. Ainsi, malheureusement, disposer d'informations incomplètes est quelque chose de fréquent et ceci peut poser problème lors de l'analyse des données, notamment des biais importants avec les algorithmes d'apprentissage automatique.

De nombreuses raisons sont à la base de ces valeurs manquantes, comme la corruption des données, la non-réponse dans le cas des sondages, l'absence d'observations mesurées par l'observateur et ainsi de suite.

Pour contourner ces problèmes, de multiples méthodes existent, comme la suppression des individus ayant une donnée manquante, ou leur substitution par des estimations.

Aujourd'hui nous disposons souvent d'énormes jeux de données pour lesquelles les données manquantes sont fréquentes et une mauvaise utilisation pourrait coûter très cher.

Nous cherchons donc des méthodes pour réduire les problèmes créés par ces valeurs manquantes, pour réduire les biais et pour augmenter la validité des conclusions d'analyses statistiques.

1.2 Problématique

Problématique : Quelle est la meilleure méthode d'imputation des données manquantes ?

1.3 Résumé

Nous présenterons les mécanismes et types des données manquantes, en parlant un peu de leur histoire, tout en donnant des exemples précis.

Ensuite, nous parlerons de quelques méthodes existantes d'imputation de données manquantes *(Imputation par moyenne/médiane, K-NN*[2]*, MICE*[3]*...)* puis des *arbres décisionnels, bootstrapping [Bre96] et forêts aléatoires (random forest)* qui seront des notions essentielles pour la compréhension de la méthode d'imputation qui nous concerne : **missing forest** [SB11].

Enfin, nous allons comparer entre elles ces méthodes sur des jeux de données avec différents types de données manquantes, afin d'étudier la performance, l'efficacité et les inconvénients de chacune.

1. Non available
2. K-Nearest Neighbours : k voisins les plus proches
3. Multiple Imputation by Chained Equations : Imputation multivariée par équations en chaîne

2 Données manquantes

2.1 Mécanismes des données manquantes

Un peu d'histoire :

— Les deux premières personnes classifiant les données manquantes sont **Little** et **Rubin**. Ils les ont présentées dans le livre **Analysis with Missing Data** en 1987 [Mis91].

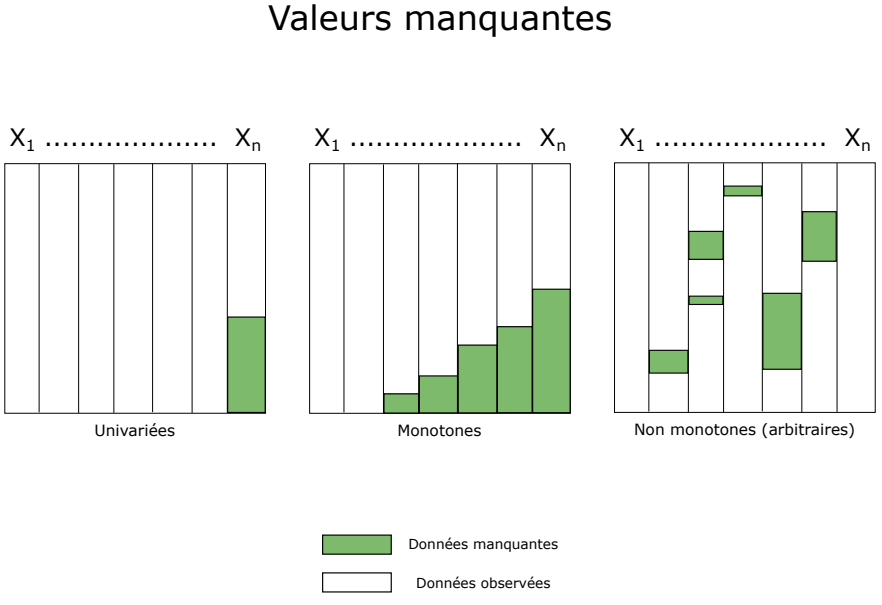
— Voici donc leur classification :
Soit $X = (x_{ij})$ un jeu de données $(n \times p)$, avec $\mathbf{X}_i = (x_{i1}, ..., x_{ip})$ la ligne de valeurs des X_j variables $(1 \leqslant j \leqslant p)$ correspondant à l'individu i. On définit la matrice d'incidence des données manquantes $\mathcal{M} = (m_{ij})$ de taille $(n \times p)$, tel que $m_{ij} = 1$ si x_{ij} est manquante et 0 sinon.

Ils considèrent trois types de répartitions de la matrice d'incidence $\mathcal{M}$, qui permettront de définir les mécanismes des données manquantes, par la suite :

1. **Les valeurs manquantes univariées :** Dans ce cas, rien de difficile, toutes les données manquantes sont isolées dans une seule variable X_k.
2. **Les valeurs manquantes monotones :** Prenons un individu i et une valeur manquante x_{ij} qui lui est attribuée. Ici, les variables qui suivent $(X_{ik}, k \geqslant j)$ seront aussi manquantes, pour l'individu i. Nous désignons un entier $M \in (1, 2, ..., p)$, comme indicateur de la première donnée manquante d'un individu. Celui-ci indique le plus petit j pour lequel x_{ij} est manquant.
3. **Les valeurs manquantes non monotones :** $\mathcal{M}$ est la matrice des valeurs manquantes $(\mathcal{M} = (m_{ij}))$. Ici, la répartition des 1 (ie : valeurs manquantes) et des 0 n'a pas de structure particulière.

Voici une représentation graphique, pour mieux comprendre :

FIGURE 2 – Types de répartition des données manquantes



Après avoir défini la matrice d'incidence des valeurs manquantes $\mathcal{M}$, on caractérise le mécanisme générant des données manquantes par la distribution conditionnelle de $\mathcal{M}$ sachant X. Cette distribution est donnée par $\mathbb{P}(M|X)$. Pour cela, on sépare les données en deux, $X_{obs} = X\mathbb{1}_{\{M=0\}}$ les données observées et $X_{mis} = X\mathbb{1}_{\{M=1\}}$, les données manquantes. Donc $X = \{X_{obs}, X_{mis}\}$

- **MCAR** *missing completely at random* : Ceci représente les données manquantes de façon complètement aléatoire, c'est-à-dire que la probabilité d'absence ne dépend pas des valeurs de X (ie. : $\mathbb{P}(M|X) = \mathbb{P}(M), \forall X$). Ce cas est peu courant dans la pratique.
- **MAR** *missing at random* : Ceci veut dire que la probabilité d'absence est liée à une ou plusieurs autres variables observées.(ie. : $\mathbb{P}(M|X) = \mathbb{P}(M|X_{obs}), \forall X_{mis}$).
- **MNAR** *missing not at random* : Ce cas est courant dans la vraie vie. La probabilité d'absence dépend de la variable en question. $\mathbb{P}(M|X)$ dépend aussi de X_{mis}.

Remarque. *Nous utiliserons pour la suite les notations ci-dessus.*

Exemple. *Nous souhaitons connaître en moyenne le nombre de livres que chaque étudiant doit rendre à la bibliothèque. Prenons chaque cas :*

- ***MCAR** : Premièrement, supposons que la BU nous donne accès à son système, où se trouve le nombre de livres à retourner pour chaque personne. Malheureusement, il y a quelques valeurs manquantes. Le gestionnaire explique donc que dans* 5% *des cas le bibliothécaire oublie d'introduire la valeur dans le système (erreur humaine). Donc ici, toutes les valeurs ont la même probabilité d'absence.*
- ***MAR** : Supposons maintenant que nous ne disposons pas de l'accès au système de la BU. Nous faisons alors un sondage anonyme auprès d'étudiants, en prenant en compte une deuxième variable (Sexe). Nous remarquons que les femmes répondent* 90% *du temps, alors que les hommes* 70%*. Dans ce cas de figure, le taux d'absence est lié à une autre variable observée.*
- ***MNAR** : Nous refaisons maintenant le sondage mais avec une variable (Nom, prénom) au lieu de (Sexe). Cette variable change fondamentalement le mécanisme car l'individu est identifiable et donc les personnes ayant plus de livres à rendre auront moins le courage de divulguer cette information. Nous pouvons, par exemple, imaginer que* 90% *des personnes ayant 0 livres répondent à la question alors que seulement* 40% *des personnes ayant plus de 5 livres répondent. Donc cet exemple illustre bien le cas des MNAR, puisque le fait d'être manquant dépend de la valeur de la variable elle-même.*

3 Méthodes existantes

Abordons maintenant les différentes méthodes de traitement de données manquantes.

Une première méthode est la **suppression pure et simple des individus où il y a des valeurs manquantes**. Nous n'allons pas aborder cette méthode, puisque elle n'est pas efficace quand les valeurs manquantes sont nombreuses.

Une autre méthode consiste dans la **substitution des valeurs manquantes d'une variable par la moyenne, ou par la médiane, mais aussi par la modalité la plus fréquente parmi les valeurs observées**.

D'autres méthodes **estiment des modèles pour prédire la valeur des données manquantes** *(Régression linéaire, maximum de vraisemblance, forêt aléatoire, etc.).*

3.1 Imputation par moyenne/médiane des valeurs

Cette méthode est l'une des premières tentatives de traiter les données manquantes. Pour des variables quantitatives, elle consiste tout simplement à remplacer la valeur manquante par la moyenne ou la médiane des valeurs observées de la colonne où elle figure. Pour le cas des variables qualitatives, l'approche est d'imputer la valeur manquante par la modalité la plus fréquente.

Voici un tableau illustrant un exemple de ce type d'imputation :

FIGURE 3 – Exemple de tableau avec valeurs manquantes

	V1	V2	V3			V1	V2	V3
Y1	7	2	2		Y1	7	2	2
Y2	2	**NA**	**NA**	→	Y2	2	**4**	**4**
Y3	**NA**	6	7	Moyenne /	Y3	**5**	6	7
Y4	8	1	**NA**	Médiane	Y4	8	1	**4**
Y5	3	7	3		Y5	3	7	3

Problème : L'imputation par moyenne ou médiane ne prend pas en considération les corrélations entre les variables, ce qui pourrait introduire un biais dans le cas des mécanismes **MAR** et **MNAR** et modifier la distribution de la variable et les intervalles de confiance associés. Ainsi, la variance de la variable est considérablement réduite, car toutes les valeurs manquantes sont remplacées par le même chiffre. Plus d'informations sur le sujet et les inconvénients de cette méthode : [Hai68]

3.2 K-NN (K nearest neighbours)

Soit une observation Y_i ayant une valeur y_{ij} manquante, une distance métrique d et un entier naturel k. La méthode **K-NN** [TCS+01] consiste à prendre les k valeurs observées ayant la distance d la plus proche de Y_i selon les autres variables $(Y_1, ..., Y_{j-1}, Y_{j+1}, ..., Y_p)$ et en faire la moyenne qui va remplacer la valeur manquante. Le choix de la distance métrique varie selon la nature du jeu de données mais n'est pas toujours évident.

Algorithme :

Soit un individu Y^* ayant une valeur manquante à la variable X_j.

1. Choisir un entier k.
2. Calculer les distances $d(Y^*, Y_i)$, pour tout $i = 1, \ldots, n$, en omettant la variable X_j.
3. Retenir les k observations $(Y_{i_1}), \ldots, (Y_{i_k})$ ayant les distances les plus petites.
4. Imputer la moyenne des valeurs des k voisins aux valeurs manquantes de l'observation Y^* :

$$y^*_{ij} = \frac{1}{k} \times ((Y_{i_1}) + \ldots + (Y_{i_k}))_j$$

Exemple. *Supposons que pour un individu, la variable salaire est manquante. Nous prenons donc les individus qui ont les mêmes caractéristiques que lui (même âge, sexe, expérience, professions, etc.). La valeur manquante sera donc imputée par la moyenne des salaires de tous ces individus.*

Remarque. *KNN rend la méthode de la moyenne un peu plus efficace. Cependant elle reste sensible aux données aberrantes. De plus, elle est très couteuse computationnellement car elle nécessite le stockage de tout*

le jeu de données. Et finalement, un bon choix de la distance d n'est pas toujours facile. Plus d'informations dans les sources : [TCS+ 01] [KN98]

3.3 Multivariate Imputation by Chained Equation (MICE)

L'imputation multiple consiste à effectuer plusieurs fois une imputation des valeurs manquantes et créer ainsi plusieurs versions complètes d'un jeu de données. Ceci est utile pour combiner des résultats et c'est plus pratique que les méthodes à une seule imputation.

Algorithme :

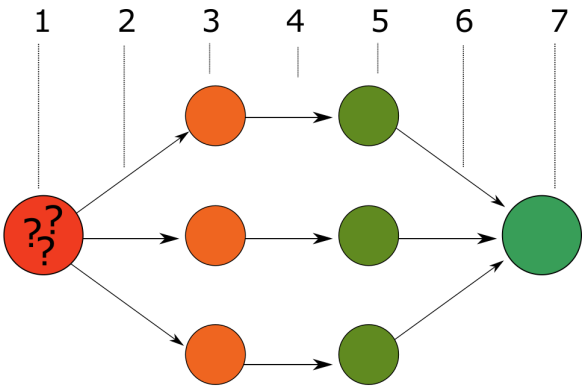
Pour $j \in J$ (ensemble des indices des variables ayant des valeurs manquantes) :

1. Première imputation (naïve) par moyenne ou médiane, par exemple.
2. Réinitialiser la première variable Y_j qui contenait des valeurs manquantes à son état de départ.
3. Faire une régression (linéaire ou autre) des données observables de la variable Y_j par rapport aux autres variables.
4. Les valeurs manquantes de Y_j sont alors remplacées par les prédictions du modèle de régression.
5. Mettre à jour le jeu de données et retourner à l'étape 2 pour le prochain j.
6. Refaire ce processus jusqu'à convergence, afin d'obtenir le résultat final de la prédiction.

Remarque. *Cette méthode est très efficace pour minimiser le biais et traiter les données quantitatives et qualitatives. Le point négatif consiste dans le fait que c'est un algorithme complexe et assez couteux. De plus, il suppose que le type de données manquantes est MAR ce qui n'est pas toujours le cas (MNAR). Plus d'informations dans la référence : [VBO99]*

Voici une représentation graphique :

FIGURE 4 – MICE



1 - Data avec valeurs manquantes
2 - Fonction mice()
3 - Data imputée
4 - Fonction with()
5 - Data analysée
6 - Fonction pool()
7 - Résultat/prédiction final(e)

3.4 Algorithme Amelia II

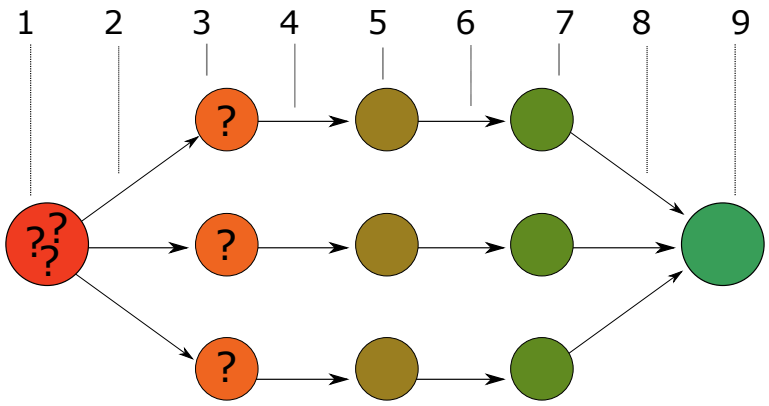
Cet algorithme a été développé par *James Honaker, Gary King et Matthew Blackwell (2001)* [HKB+]. C'est une méthode qui s'appuie sur l'inférence bayésienne combinant l'algorithme **EM** (maximum de vraisemblance) avec une approche bootstrap [4] [Bre96] et sur l'hypothèse de normalité $X \sim N_k(\mu, \sigma)$, où X est le tableau des données original et $(\mu, \sigma) = \theta$ les paramètres du modèle.

Une autre hypothèse est que les données sont MAR, ie : $p(M|X) = p(M|X_{obs})$, avec X_{obs} la partie observée du jeu de données.

L'algorithme est donc une méthode d'imputation multiple, il utilise l'algorithme EM sur différents échantillons bootstrap du jeu de données, afin de trouver des valeurs des paramètres du jeu de données complet de la loi à posteriori $p(\theta|X_{obs})$ [HKB+].

Les imputations sont alors créées par tirage de X_{mis} selon sa distribution conditionnelle sur X_{obs} et des tirages de θ. Nous obtenons alors m "valeurs candidates" pour chaque valeur manquante, et donc m jeu de données complet que Amelia va moyenner, pour obtenir la meilleure prédiction.

FIGURE 5 – Algorithme Amelia II



1 - Data avec valeurs manquantes
2 - Echantillonage par bootstrap
3 - Sous-ensembles par bootstrap
4 - Algorithme du maximum de vraisemblance (EM)
5 - Jeu de donnée imputé
6 - Analyse statistique
7 - Résultats séparés
8 - Moyenner les résultats
9 - Résultat/prédiction final(e)

4. Rééchantillonnage

4 MissForest imputation

4.1 Arbre décisionnel

Définition (Arbre)**.** *En théorie des graphes, un arbre est un graphe ayant les trois propriétés suivantes :*

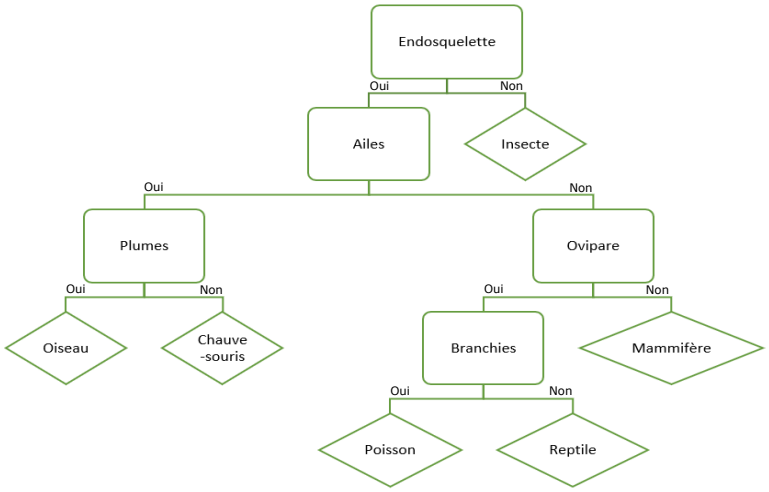
- ***Non orienté*** *: Le graphe est composé de sommets et d'arêtes, tel que chaque arête est une paire de sommets distincts (plutôt que des couples comme pour le graphe orienté). Elles ne possèdent pas de direction particulière .*
- ***Acyclique*** *: (Ne contenant aucun cycle) C'est-à-dire qu'il ne contient pas de chaîne (fermée) d'arêtes consécutives, dont les deux sommets d'extrémités sont identiques.*
- ***Connexe*** *: Pour toutes les paires de sommet (par exemple le sommet u et v), il existe une chaîne les reliant.*

Un arbre commence par un nœud principal appelé "Nœud Racine", puis des "Nœuds internes" (nœuds ayant des ascendants et des descendants) et enfin des " Nœuds terminaux " ou " feuilles " qui ont des ascendants mais pas de descendants.

Définition (Arbre décisionnel)**.** *Les arbres décisionnels sont une catégorie d'arbres utilisée dans les domaines d'aide à la décision (l'informatique, la sécurité, l'exploration de données...) modélisant une hiérarchie de tests pour prédire un résultat. Ils permettent d'évaluer les différentes décisions possibles en fonction de leur probabilité, leur bénéfice et leur coût, en parcourant les " nœuds internes ". Ces nœuds permettent de tester les attributs (représentation catégorielle des données), jusqu'à ce qu'on arrive à une décision finale (feuille).*

Dû à la sensibilité des arbres décisionnels aux bruits et aux points aberrants, une autre utilisation en Machine Learning de ces graphes consiste à construire une « Forêt d'arbres décisionnels » ou « Forêt aléatoire ».

FIGURE 6 – Exemple d'arbre décisionnel



4.2 Bootstrap aggregating - Forêt aléatoire (Random Forest)

L'algorithme de forêt aléatoire (ensemble d'arbres décisionnels) fut introduit pour la première fois par *Tin Kam Ho (1995) [Tin95], puis formellement par Leo Breiman (2001) et Adèle Cutler* [Bre01]. Cet algorithme consiste à utiliser le principe du « bagging » [5] [Bre96] et des sous-espaces aléatoires [6], afin d'améliorer la classification d'un arbre de décision, en réduisant la variance de l'estimateur, sans pour autant augmenter le biais.

Le principe est de créer de multiples arbres décisionnels à partir de différents échantillons « bootstrap » et différentes variables de segmentation (choisies aléatoirement). Les résultats des différents arbres sont alors moyennés (cas de données quantitatives) ou utilisés pour un vote de majorité (cas de données qualitatives), pour en tirer enfin la meilleure prédiction.

Exemple. *(Analogie) Imaginons qu'un nouvel élève à Montpellier veut aller à un restaurant. Il demande alors à un de ses collègues de classe de lui donner quelques recommandations. Son camarade lui pose quelques questions sur ses préférences gastronomiques, son budget et ainsi de suite. Il lui propose donc un restaurant par rapport à ses réponses.*

Ceci est un exemple typique du fonctionnement d'un arbre décisionnel. Imaginons maintenant que notre élève demande à toute sa promotion de lui recommander un restaurant. Encore une fois, chaque collègue lui en propose un, en se basant sur ses réponses. Finalement, il choisit le restaurant qui a reçu le plus de recommandations. Cette approche est celle d'une forêt aléatoire.

4.2.1 Algorithme

Soit un jeu de données $X = (X_1, ..., X_n)$

Pour $b = 1, ..., B$:

1. Tirer un échantillon de taille n avec remise de X, appelé X_b *(échantillon bootstrap)*.
2. Construire (entrainer) un arbre décisionnel t_b sur X_b.

Après l'entrainement, les prédictions des échantillons x' pour les données manquantes peuvent être faites en moyennant les prédictions des données quantitatives sur x' ($\hat{f} = \frac{1}{B}\sum_{b=1}^{B} t_b(x')$), ou en prenant la modalité majoritaire pour les données qualitatives.

5. Bootstrap AGGregatING - le principe est de créer de nouveaux échantillons au hasard par tirage avec remise d'observations
6. Sous-ensembles aléatoires des variables formées par projections aléatoires

4.3 MissForest algorithm

Développé par *Stekhoven et Bühlmann (2011)* [SB11], **missforest** est une technique itérative s'appuyant sur les forêts aléatoires, pour faire une imputation multiple. Il consiste à entraîner une forêt aléatoire sur chaque variable contenant des valeurs manquantes, pour prédire ces valeurs.

Les avantages de ce package c'est le fait qu'il dispose en lui l'erreur **OOB**[7] *(Out Of the Bag)* [Bre], afin d'estimer l'erreur de l'imputation, sans utiliser un jeu de données test.

De plus, l'algorithme peut gérer les **mixed-type data**[8], les interactions complexes entre les variables, les dimensions très grandes ($p >> n$) et ne nécessite aucune préparation ou transformation du jeu de données. Noter que cet algorithme est le seul à notre connaissance permettant de traiter des données mixtes *(quantitatives et qualitatives)*, qui sont très fréquentes dans les jeux de données modernes.

4.3.1 Algorithme

Notations :

Soit une variable Y^s ayant des valeurs manquantes indexées par $i^s_{mis} \subseteq 1, ..., n$. On définit :

1. y^s_{obs} les valeurs observées dans Y^s.
2. y^s_{mis} les valeurs manquantes dans Y^s.
3. $X^s = Y \backslash Y^s$ l'ensemble des régresseurs de Y^s parmi lesquels on considère : x^s_{obs} les régresseurs observés pour $i^s_{obs} = \{i, ..., n\} \backslash i^s_{mis}$ et x^s_{mis} les régresseurs manquants pour i^s_{mis}.
   La méthode suit alors l'algorithme suivant : Soit X une matrice $n \times p$ et gamma un critère d'arrêt.
   (a) Faire une première imputation "naïve" des valeurs manquantes. *(ex : Moyenne)*
   (b) $K \leftarrow$ vecteur des indices de colonnes de X triées par ordre croissant suivant la quantité de valeurs manquantes
   (c) Tant que gamma n'est pas atteint :
      i. Y^{old}_{imp} la matrice précédemment imputée.
      ii. Pour s dans K faire :
         A. Ajuster $y^s_{obs} \sim x^s_{obs}$ par forêt aléatoire.
         B. Prédire y^s_{mis} avec les régresseurs x^s_{mis}
         C. Y^{new}_{imput} nouvelle matrice imputé par les prédictions y^s_{mis}.
      iii. Mettre à jour le critère γ.
   (d) Retournez la matrice finale X^{imp}.

7. En machine learning, l'OOB error est une méthode mesurant l'erreur de prédiction d'un modèle utilisant le bootstrap aggregating (ex : Forêt aléatoire)
8. Jeu de données contenant des variables quantitatives et qualitatives

5 Analyse et comparaison

5.1 Imputation avec toutes les méthodes pour différents types de data sets

Protocole :

- Choisir un jeu de données.
- Compiler les codes pour chaque type : **MCAR, MAR** et **MNAR**.
- Générer pour chacun des trois mécanismes, décrits en haut, des jeux de données avec 10%, 20% et 40% de valeurs manquantes.
- Créer des graphiques, pour une meilleure représentation du problème.
- Regrouper les erreurs d'imputation dans un **tableau**, pour ensuite les comparer.
- Choisir la meilleure méthode d'imputation.

Structure :

- **MCAR**, **MNAR**, **MAR**
  - 10%, 20%, 40%
    - KNN imputation
    - Missforest imputation
    - MICE
    - Amelia

Les packages à utiliser se trouvent dans la partie **Annexe**, ainsi qu'une partie des codes.

Nous choisissons un jeu de données *(code en annexe)* sur le site de *UCI repository*[9]. Il est constitué de 270 observations et 14 variables mixtes *(quantitatives et qualitatives)*.

Ce jeu de données comprend les variables suivantes :

- âge
- sex : 0 = Femme, 1 = Homme
- cp (chest pain) : douleur au niveau de la poitrine de 1(plus forte) à 4(moins forte)
- pression : pression artérielle au repos (en mmHG)
- chol : cholestérol (en mg/dl)
- fbs : glycémie à jeun
- restecg : Électrocardiographie au repos
  - 1 = Normal
  - 2 = Anormal
  - 3 = Hypertrophie ventriculaire gauche
- maxrate : maximum de fréquence cardiaque
- exang : angine de poitrine dû aux exercices (1 = oui, 0 = non)
- oldpeak : Dépression segment ST dû à l'exercice.
- slope : tangente du segment ST à l'exercice.
  - 1 = croissante
  - 2 = constante
  - 3 = décroissante
- vessels : nombre de vaisseaux sanguins majeurs (0-3) coloré au fluor
- thal : (thallium heart scan) Scintigraphie au Thallium
  - 3 = normale (pas de zone froide)
  - 6 = zones froides au repos et à l'exercice
  - 7 = zones froides que pendant l'exercice
- hd (heart disease) : 1 = Sain, 2 = Malade

9. Statlog (heart) Data Set, http ://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/

Voici un aperçu de quelques observations du jeu de données (qualitatives, quantitatives, modalités...) :

FIGURE 7 – Quelques observations

```
> head(heart)    # Aperçu de quelques observations
  age sex cp pression chol fbs restecg maxrate exang oldpeak slope vessels thal hd
1  70   1  4      130  322   0       2     109     0     2.4     2       3    3  2
2  67   0  3      115  564   0       2     160     0     1.6     2       0    7  1
3  57   1  2      124  261   0       0     141     0     0.3     1       0    7  2
4  64   1  4      128  263   0       0     105     1     0.2     2       1    7  1
5  74   0  2      120  269   0       2     121     1     0.2     1       1    3  1
6  65   1  4      120  177   0       0     140     0     0.4     1       0    7  1
```

D'après la description du jeu de données sur l'UCI repository, les variables *age, cp, fbs, restecg, exang et slope* sont *qualitatives (factors)*. En regardant les sorties R ci-dessus, on observe que ces variables sont présentées sous forme numérique. On utilise donc $as.factor()$ pour rendre ces variables qualitatives.

FIGURE 8 – Structure corrigée

```
> str(heart)
'data.frame':   270 obs. of  14 variables:
 $ age     : num  70 67 57 64 74 65 56 59 60 63 ...
 $ sex     : Factor w/ 2 levels "0","1": 2 1 2 2 1 2 2 2 2 1 ...
 $ cp      : Factor w/ 4 levels "1","2","3","4": 4 3 2 4 2 4 3 4 4 4 ...
 $ pression: num  130 115 124 128 120 120 130 110 140 150 ...
 $ chol    : num  322 564 261 263 269 177 256 239 293 407 ...
 $ fbs     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...
 $ restecg : Factor w/ 3 levels "0","1","2": 3 3 1 1 3 1 3 3 3 3 ...
 $ maxrate : num  109 160 141 105 121 140 142 142 170 154 ...
 $ exang   : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 2 2 1 1 ...
 $ oldpeak : num  2.4 1.6 0.3 0.2 0.2 0.4 0.6 1.2 1.2 4 ...
 $ slope   : Factor w/ 3 levels "1","2","3": 2 2 1 2 1 1 2 2 2 2 ...
 $ vessels : Factor w/ 4 levels "0","1","2","3": 4 1 1 2 2 1 2 2 3 4 ...
 $ thal    : Factor w/ 3 levels "3","6","7": 1 3 3 3 1 3 2 3 3 3 ...
 $ hd      : Factor w/ 2 levels "1","2": 2 1 2 1 1 1 2 2 2 2 ...
```

5.1.1 Imputation des données manquantes MCAR

10% de données manquantes

Nous allons mettre les codes pour les 10% des valeurs manquantes seulement. Pour les autres 20% et 40%, ainsi que pour les parties de MNAR et MAR, ceux-ci se trouveront dans l'Annexe. Les résultats (erreurs d'imputation pour chaque méthode) ne seront pas affichés dans ce paragraphe, mais plutôt regroupés en tableau comparatif au paragraphe 6 : Résultats et comparaison.

FIGURE 9 – Nombre de valeurs manquantes dans chaque variable - 10%

```
> summary(heart.train)  #On voit ici le nombre de valeurs manquantes pour chaque variable.
      age             sex          cp          pression          chol            fbs          restecg          maxrate          exang          oldpeak
 Min.   :29.00   0    : 87   1: 20   Min.   : 94.0   Min.   :126.0   0   :229   0   :130   Min.   : 71.0   0   :180   Min.   :0.000
 1st Qu.:48.00   1    :179   2: 42   1st Qu.:120.0   1st Qu.:213.0   1   : 40   1   :  2   1st Qu.:132.5   1   : 89   1st Qu.:0.000
 Median :55.00   NA's:  4   3: 79   Median :130.0   Median :245.0   NA's:  1   2   :135   Median :153.0   NA's:  1   Median :0.800
 Mean   :54.43              4:129   Mean   :131.4   Mean   :248.5              NA's:  3   Mean   :149.5              Mean   :1.054
 3rd Qu.:61.00                      3rd Qu.:140.0   3rd Qu.:277.0                         3rd Qu.:166.0              3rd Qu.:1.650
 Max.   :77.00                      Max.   :200.0   Max.   :417.0                         Max.   :202.0              Max.   :6.200
                                    NA's   :1       NA's   :1                             NA's   :3                  NA's   :2
  slope      vessels        thal          hd
 1   :129   0   :159   3   :152   1   :146
 2   :121   1   : 58   6   : 14   2   :118
 3   : 18   2   : 33   7   :102   NA's:  6
 NA's:  2   3   : 19   NA's:  2
            NA's:  1
```

Remarque. *La sortie de code ci-dessous donne, pour chaque variable, le minimum, le maximum, la médiane, la moyenne, le premier et troisième quantile et le nombre de valeurs manquantes.*

Pour la suite, nous allons tester les différentes méthodes d'imputation. Pour chacune de ces méthodes, nous calculons son erreur (affichée dans le tableau des comparaisons des erreurs), qui va nous servir à la comparaison finale des résultats. Voici donc les codes, sans sorties :

KNN imputation

```
heart.knn10 <- kNN(heart.mcar10, k= 10 ,imp_var = FALSE)

# Calcul de l'erreur
err.knn10 <- mixError(heart.knn10, heart.mcar10, data.matrix(heart))
```

Missforest imputation

```
heart.missforest10<-missForest(heart.mcar10, maxiter=10,
                               ntree = 200)

heart.missforest10$OOBerror
# L'erreur NRMSE pour Missforest est integree dans le package
# donc il est inutile de la calculer comme ce qu'on a fait pour
# les autres methodes.
```

MICE

```
heart.mice10 <- mice(heart.mcar10, m = 1)  # Imputation multiple

heart.mice.complet10 <- complete(heart.mice10)
# On combine les valeurs imputees en un jeu de donnees

err.mice10 <- mixError(heart.mice.complet10, heart.mcar10, data.matrix(heart))
```

Amelia

```
heart.amelia10 <- amelia(heart.mcar10 , m=5, noms = c(2,3,6,7,9,11,12,13,14))

err.amelia10 <- mixError(heart.amelia10$imputations$imp1,
                                    heart.mcar10, data.matrix(heart))
```

Nous avons effectué les mêmes opérations pour 20% et 30% des valeurs manquantes.

Les codes pour **MNAR** et **MAR** se trouvent également dans l'annexe.

20% **de données manquantes**

FIGURE 10 – Nombre de données manquantes pour chaque variable - 20%

```
> summary(heart.train)
      age          sex         cp          pression          chol          fbs       restecg          maxrate        exang          oldpeak        slope
 Min.   :29.00   0   : 87   1   : 20   Min.   : 94.0   Min.   :126.0   0   :225   0   :129   Min.   : 71.0   0   :177   Min.   :0.000   1   :128
 1st Qu.:47.50   1   :179   2   : 42   1st Qu.:120.0   1st Qu.:212.5   1   : 40   1   :  2   1st Qu.:132.5   1   : 89   1st Qu.:0.000   2   :120
 Median :55.00   NA's:  4   3   : 79   Median :130.0   Median :245.0   NA's:  5   2   :134   Median :153.0   NA's:  4   Median :0.800   3   : 18
 Mean   :54.43              4   :128   Mean   :131.4   Mean   :248.6              NA's:  5   Mean   :149.5              Mean   :1.054   NA's:  4
 3rd Qu.:61.00              NA's:  1   3rd Qu.:140.0   3rd Qu.:279.0                         3rd Qu.:166.0              3rd Qu.:1.700
 Max.   :77.00                         Max.   :200.0   Max.   :417.0                         Max.   :202.0              Max.   :6.200
 NA's   :3                             NA's   :1       NA's   :3                             NA's   :3                  NA's   :3
 vessels      thal        hd
 0   :157   3   :150   1   :143
 1   : 57   6   : 14   2   :117
 2   : 33   7   :102   NA's: 10
 3   : 19   NA's:  4
 NA's:  4
```

40% **de données manquantes**

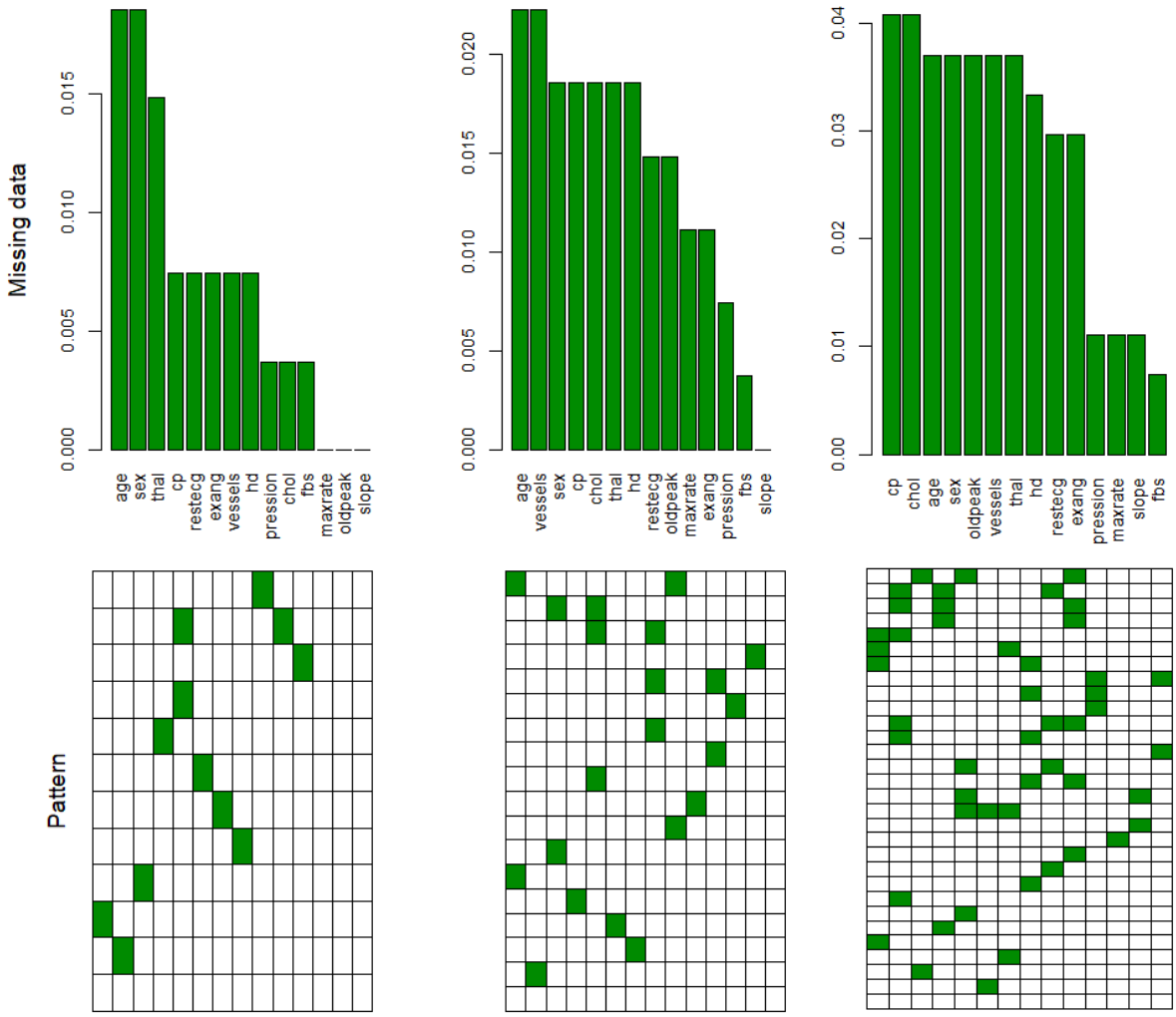
FIGURE 11 – Nombre de données manquantes pour chaque variable - 40%

```
> summary(heart.train)
      age          sex         cp          pression          chol          fbs       restecg          maxrate        exang          oldpeak
 Min.   :29.00   0   : 87   1   : 19   Min.   : 94.0   Min.   :126.0   0   :223   0   :129   Min.   : 71.0   0   :175   Min.   :0.00
 1st Qu.:47.00   1   :177   2   : 41   1st Qu.:120.0   1st Qu.:213.0   1   : 40   1   :  2   1st Qu.:132.0   1   : 88   1st Qu.:0.00
 Median :55.00   NA's:  6   3   : 76   Median :130.0   Median :245.0   NA's:  7   2   :133   Median :153.0   NA's:  7   Median :0.80
 Mean   :54.23              4   :125   Mean   :131.3   Mean   :249.2              NA's:  6   Mean   :149.4              Mean   :1.03
 3rd Qu.:61.00              NA's:  9   3rd Qu.:140.0   3rd Qu.:281.2                         3rd Qu.:166.0              3rd Qu.:1.60
 Max.   :77.00                         Max.   :200.0   Max.   :417.0                         Max.   :202.0              Max.   :5.60
 NA's   :8                             NA's   :5       NA's   :6                             NA's   :9                  NA's   :9
  slope      vessels      thal        hd
 1   :125   0   :157   3   :149   1   :142
 2   :118   1   : 56   6   : 14   2   :115
 3   : 17   2   : 33   7   :100   NA's: 13
 NA's: 10   3   : 18   NA's:  7
            NA's:  6
```

```
# Visualisation valeurs manquantes
heart.mcar10.plot <- aggr(heart.mcar10, col=c('white','green4'),
numbers=TRUE, sortVars=TRUE,
labels=names(heart.mcar10), cex.axis=.9,
gap=3, ylab=c("Missing data","Pattern"))

# La meme chose pour 20% et 40%
```

FIGURE 12 – Mécanisme MCAR pour différents pourcentages

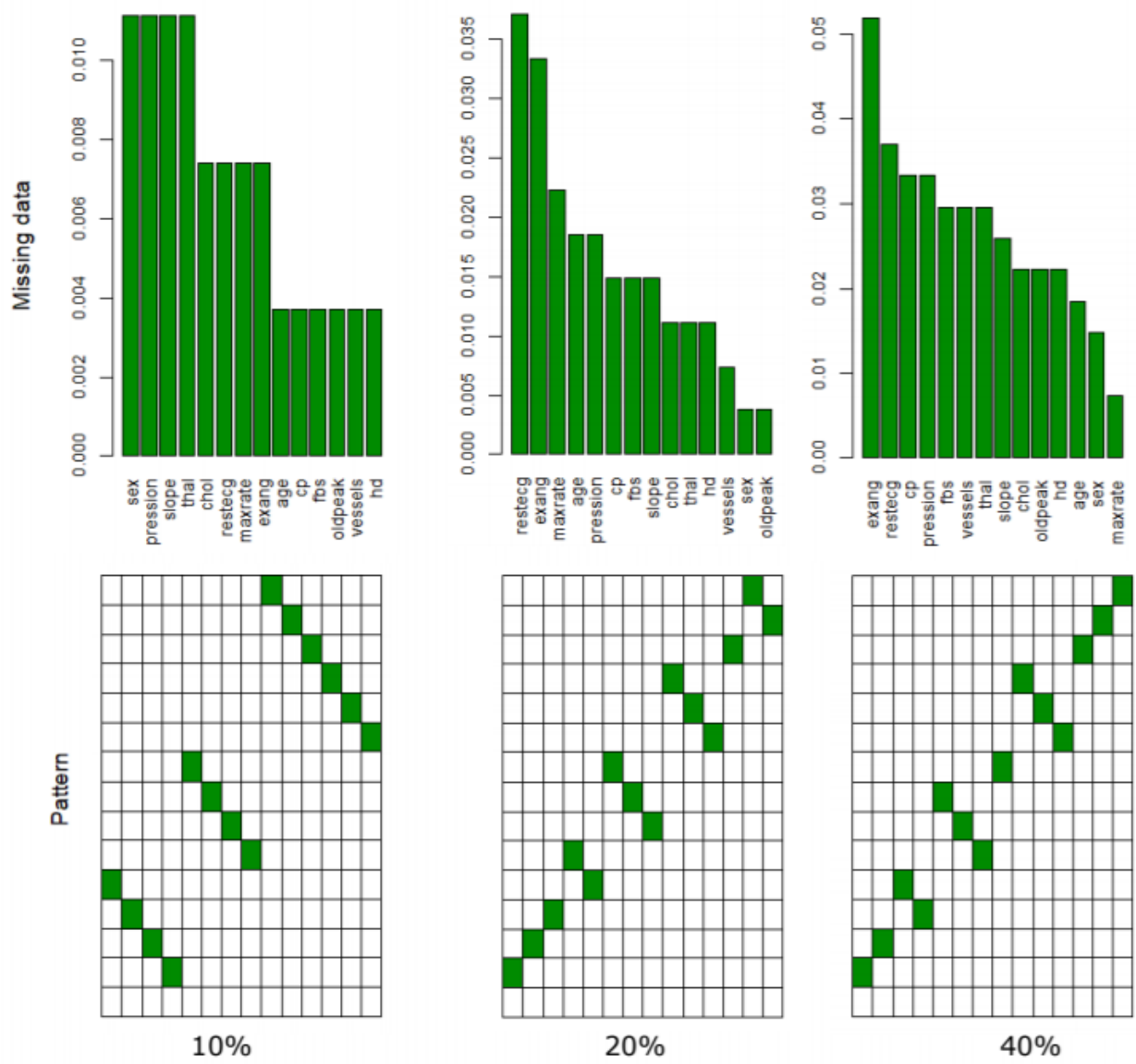


5.1.2 Imputation des données manquantes MAR

```
# Visualisation valeurs manquantes
heart.mar10.plot <- aggr(heart.mar10, col=c('white','green4'),
                         numbers=TRUE, sortVars=TRUE,
                         labels=names(heart.mar10), cex.axis=.9,
                         gap=3, ylab=c("Missing data","Pattern"))

# La meme chose pour 20% et 40%
```

FIGURE 13 – Mécanisme MAR pour différents pourcentages

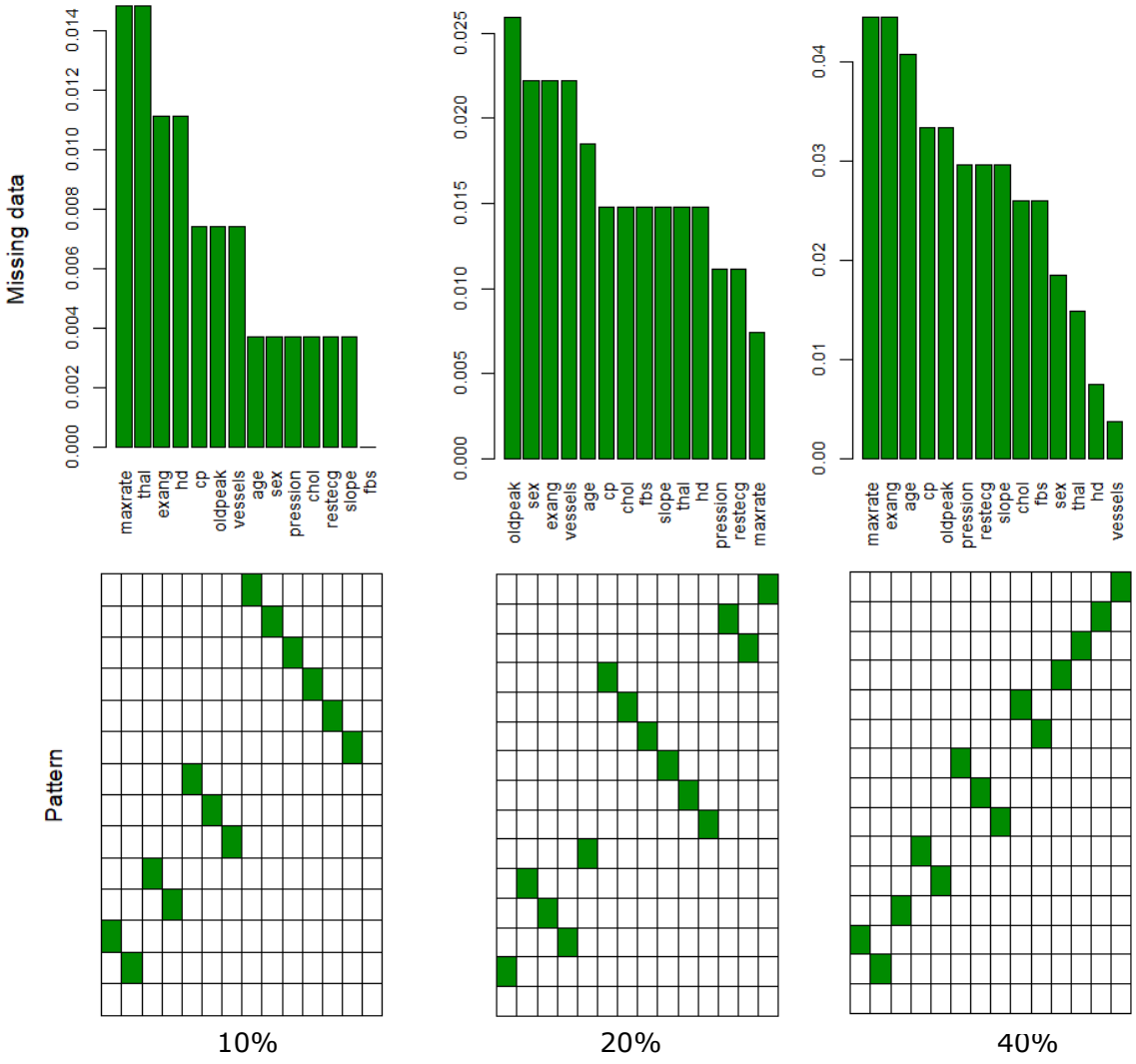


5.1.3 Imputation des données manquantes MNAR

```
# Visualisation valeurs manquantes
heart.mnar10.plot <- aggr(heart.mnar10, col=c('white','green4'),
                          numbers=TRUE, sortVars=TRUE,
                          labels=names(heart.mnar10), cex.axis=.9,
                          gap=3, ylab=c("Missing data","Pattern"))

# La meme chose pour 20% et 40%
```

FIGURE 14 – Mécanisme MNAR pour différents pourcentages



Les figures ci-dessus montrent le pourcentage de données manquantes pour chaque variable, ainsi que les différentes distributions des valeurs manquantes, selon le type de mécanisme et le pourcentage de données manquantes. On voit bien que le premier graphique **(MCAR)** représente une distribution aléatoire des données manquantes, alors que les deux autres graphiques montrent une structure particulière *(expliqué dans la partie 2.1)*.

6 Résultats et interprétation

Nous allons comparer, dans cette section, les résultats des simulations. La comparaison est basée sur NRMSE *(normalized root-mean-square error)* qui est l'erreur d'imputation des valeurs manquantes. Elle est obtenue par la formule suivante :

$$NRMSE = \sqrt{\frac{E((X - X*)^2)}{Var(X)}}$$

Avec : X le jeu de données original et $X*$ le jeu de données imputé.

FIGURE 15 – Tableau de comparaison des erreurs NRMSE

Pour MCAR :

Pourcentage Imputation	10%	20%	40%
KNN	13.10175	39.32759	29.60917
MissForest	**10.41169**	**37.32926**	**28.01442**
MICE	33.22436	49.57206	49.54468
AMELIA II	24.44788	42.6654	60.13478

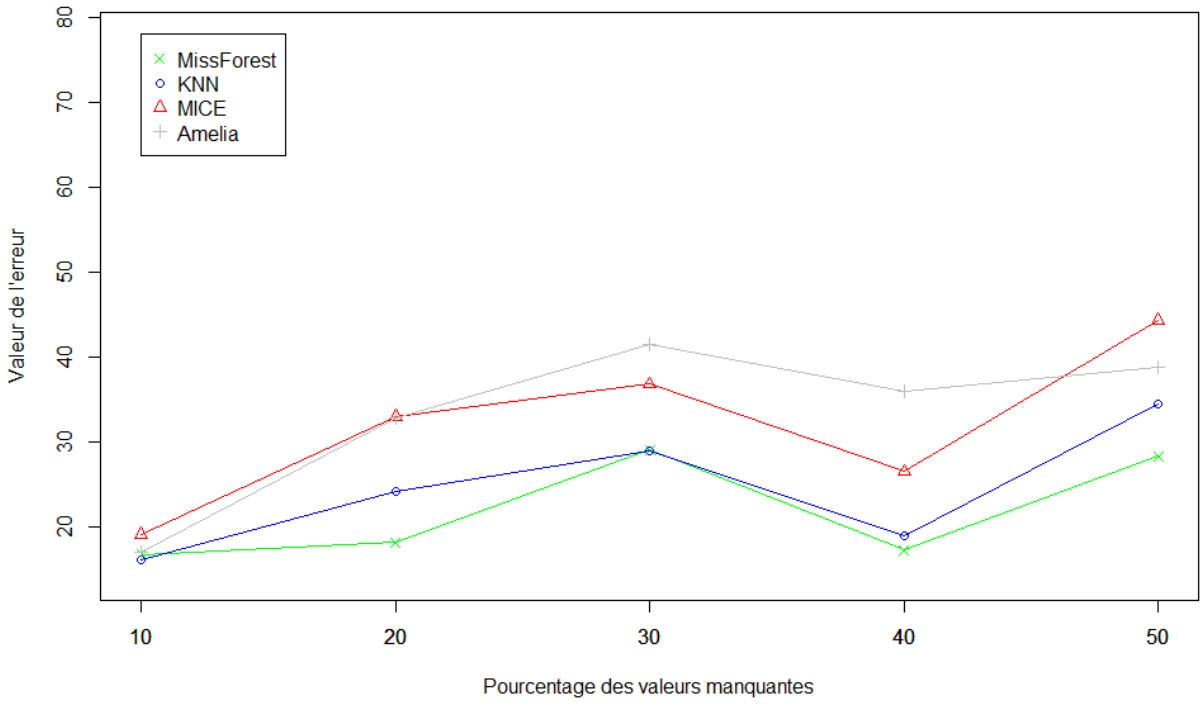
Pour MAR :

Pourcentage Imputation	10%	20%	40%
KNN	12.68164	18.44051	17.42983
MissForest	**12.97294**	**17.55679**	**15.33786**
MICE	23.64318	26.09572	36.73166
AMELIA II	17.34554	21.43214	29.4989

Pour MNAR :

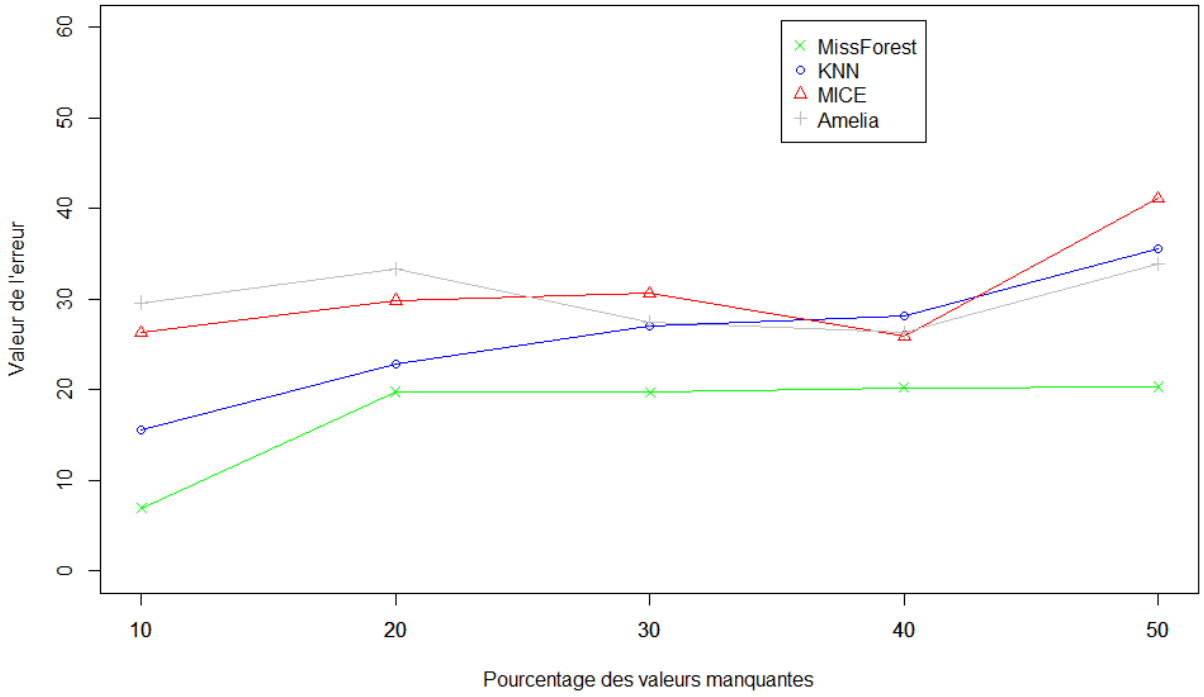
Pourcentage Imputation	10%	20%	40%
KNN	26.6655	32.30078	48.56432
MissForest	**19.21056**	**20.71287**	**14.94464**
MICE	28.56685	35.51513	43.07743
AMELIA II	25.59228	36.03463	42.75454

FIGURE 16 – Evolution des erreurs d'imputation par rapport à la proportion des données manquantes en MCAR



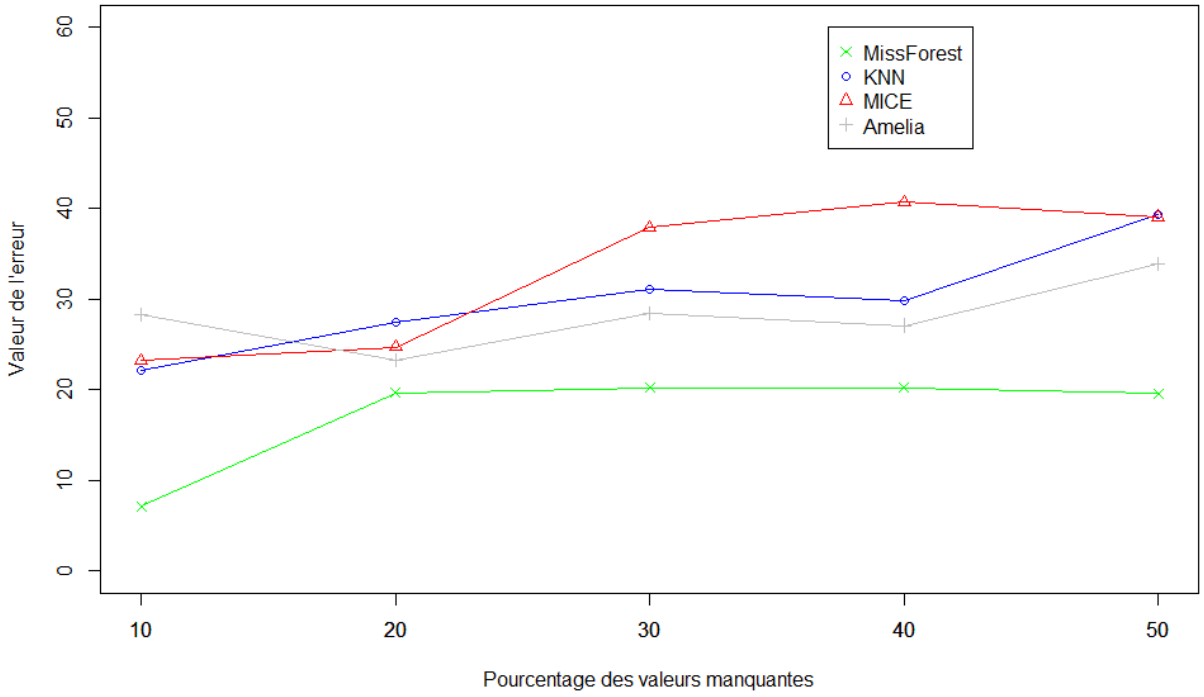
Pour les données MCAR : La différence entre les erreurs **NRMSE** n'est pas significative à 10%. Mais l'augmentation des valeurs manquantes , notamment à 50%, induit une plus grande différence de NRMSE pour **Amelia** et **MICE**.

FIGURE 17 – Evolution des erreurs d'imputation par rapport à la proportion des données manquantes en MAR



Pour les données MAR : Ici, l'algorithme **Missforest** se distingue plus des autres méthodes. En effet, nous remarquons que leur erreur d'imputation est plus sensible quand la proportion de données manquantes est grande, contrairement à **Missforest** qui n'évolue pas entre 20% et 40%.

FIGURE 18 – Evolution des erreurs d'imputation par rapport à la proportion des données manquantes en MNAR



Pour les données MNAR : Les erreurs d'imputation de **KNN**, **Mice** et **Amelia II** sont plus grandes pour le type **MNAR** et augmentent avec le pourcentage de valeurs manquantes. L'algorithme **Missforest** se distingue largement pour ce type de données manquantes, puisque son NRMSE est 2 fois plus petit que les NRMSE des 3 autres algorithmes.

7 Conclusion

Avec ceci, nous pouvons conclure que **MissForest** est une méthode d'imputation optimale, pour un mixted-type data set et elle est meilleure que d'autres méthodes d'imputation, comme **KNN, AMELIA II** et **MICE**.

Elle se distingue quand la proportion des valeurs manquantes augmente. De plus, elle offre de meilleurs résultats pour les mécanismes de données manquantes **MAR** et **MNAR**.

Il faut pourtant noter que l'erreur **NRMSE** n'est pas la meilleure méthode de comparaison. Cependant, c'est la seule manière proposée par la littérature, pour l'instant [AC92] [HK06] [WM06].

Références

[AC92] J.Scott Armstrong and Fred Collopy. Error measures for generalizing about forecasting methods : Empirical comparisons. International Journal of Forecasting, 8(1) :69 – 80, 1992. 22

[Bre] Leo Breiman. Out-of-bag estimation. Statistics Department, University of California, Berkeley, CA. 94708. 11

[Bre96] Leo Breiman. Bagging predictors. Machine learning, 24(2) :123–140, 1996. 3, 8, 10

[Bre01] Leo Breiman. Random forests. Machine learning, 45(1) :5–32, 2001. 10

[Hai68] Yoel Haitovsky. Missing data in regression analysis. Journal of the Royal Statistical Society. Series B (Methodological), 30(1) :67–82, 1968. 6

[HK06] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. International journal of forecasting, 22(4) :679–688, 2006. 22

[HKB+] James Honaker, Gary King, Matthew Blackwell, et al. Amelia ii : A program for missing data. 8

[KN98] Edwin M Knox and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. In Proceedings of the international conference on very large data bases, pages 392–403. Citeseer, 1998. 7

[Mis91] Robert J. Mislevy. Journal of Educational Statistics, 16(2) :150–155, 1991. 4

[SB11] Daniel J. Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics, 28(1) :112–118, 10 2011. 3, 11

[TCS+01] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. Bioinformatics, 17(6) :520–525, 2001. 6, 7

[Tin95] Tin Kam Ho. Random decision forests. In Proceedings of 3rd International Conference on Document Analysis and Recognition, volume 1, pages 278–282 vol.1, Aug 1995. 10

[VBO99] Stef Van Buuren and Karin Oudshoorn. Flexible multivariate imputation by MICE. 1999. 7

[WM06] Cort J. Willmott and Kenji Matsuura. On the use of dimensioned measures of error to evaluate the performance of spatial interpolators. International Journal of Geographical Information Science, 20 :89–102, 2006. 22

8 Sources

— Pages web :
  — http://archive.ics.uci.edu/ml/datasets/statlog+(heart)
  — https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-idm.pdf
  — https://academic.oup.com/bioinformatics/article/28/1/112/219101#2447858
  — https://books.google.fr/booksid=PaiODwAAQBAJ&printsec=frontcover&hl=fr&source=gbs_ViewAPI&redir_esc=y#v=onepage&q&f=true
  — https://www.dataanalyticspost.com/
  — https://fr.wikiversity.org/wiki/Arbres_de_d%C3%A9cision/D%C3%A9finition
  — https://www.dataapplab.com/python_missing_data/
  — https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/#R23
  — https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/

— Figures crées avec *Inkscape* et adaptées à partir des sources suivantes :
  — https://www.kisspng.com/png-missing-data-diagram-information-imputation-market-1883539/
  — https://gking.harvard.edu/files/gking/files/amelia_jss.pdf
  — https://slideplayer.com/slide/6895721/
  — http://callisto.ggsrv.com/imgsrv/FastFetch/UBER1/ZI-5EEE-2015-ANN00-IDSI-12146-1
  — Marcelino et al., Improved Methods for the Imputation of Missing Data in Pavement Management Systems, In : 18th Annual International Conference on Pavement Engineering, Asphalt Technology and Infrastructure, Liverpool, 2019.
  — Buuren and Groothuis-Oudshoorn, 2010

— Articles et livres
  — Ho, Tin Kam (1995). Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.
  — Breiman L (2001). "Random Forests". Machine Learning. 45 (1) : 5–32.
  — Liaw A (16 October 2012). "Documentation for R package randomForest" (PDF). Retrieved 15 March2013.
  — Raghunathan TW, Lepkowksi JM, Van Hoewyk J, Solenbeger P. A multivariate technique for multiply imputing missing values using a sequence of regression models. Survey Methodology. 2001 ;27 :85–95.
  — Van Buuren S. Multiple imputation of discrete and continuous data by fully conditional specification. Stat Methods Med Res. 2007 Jun ; 16(3) :219-42.
  — Honaker J., King G. et Blackwell M., Amelia II : A Program for Missing Data, Journal of statistical software 45 (2011), no 7.
  — Stekhoven, D.J. and Bühlmann, P., 2011. MissForest—non-parametricmissing value imputation for mixed-type data. Bioinformatics, 28(1), pp.112-118

9 Annexe

Structure - annexe :

- Packages à installer
- Préparation et visualisation du jeu de données
- **MCAR**, **MNAR**, **MAR**
  - 10%, 20%, 40%
    - KNN imputation
    - Missforest imputation
    - MICE
    - Amelia

9.1 Codes

Voici les packages à installer :

```
install.packages("VIM")
install.packages("hydroGOF")
install.packages("Hmisc")
install.packages("Amelia")
install.packages("mice")
install.packages("missForest")

library(missForest)
library(mice)
library(VIM)
library(hydroGOF)
library(Hmisc)
library(Amelia)
```

La préparation du jeu des données et leur visualisation :

```
heart <- read.table("D:/missforest code/heart-data/heart.dat", sep = " " )

colnames(heart) <- c("age", "sex", "cp", "pression", "chol", "fbs", "restecg",
                     "maxrate", "exang", "oldpeak", "slope", "vessels",
                     "thal","hd")

# Visualisation des premieres valeurs
head(heart)
```

Correction de la structure du jeu de données :

```
heart$sex <- as.factor(heart$sex)
heart$cp <- as.factor(heart$cp)
heart$fbs <- as.factor(heart$fbs)
heart$restecg <- as.factor(heart$restecg)
heart$exang <- as.factor(heart$exang)
heart$slope <- as.factor(heart$slope)
heart$vessels <- as.integer(heart$vessels)
heart$vessels <- as.factor(heart$vessels)
heart$thal <- as.integer(heart$thal)
heart$thal <- as.factor(heart$thal)
heart$hd <- as.factor(heart$hd)
```

```
# Ratio des donnees manquantes
test.ratio <- 0.1

# Indices de l'echantillon test
IND <- which(!is.na(heart),arr.ind=TRUE)

ntest <- ceiling(nrow(heart) * test.ratio)
ind.test <- IND[sample(1:dim(IND)[1],ntest),]

# Creation des donnees manquantes
heart.test <- heart[ind.test]

# Data set avec les valeurs manquantes
heart.mcar10 <- heart
heart.mcar10[ind.test] <- NA
```

9.1.1 MCAR

Code 20% données manquantes

```
# Ratio de donnees manquantes
test.ratio <- 0.2

# Indices de l'echantillon test
IND <- which(!is.na(heart),arr.ind=TRUE)
ntest <- ceiling(nrow(heart) * test.ratio)
ind.test <- IND[sample(1:dim(IND)[1],ntest),]

# Creation des donnees manquantes
heart.test <- heart[ind.test]
heart.mcar20 <- heart
heart.mcar20[ind.test] <- NA

# Visualisation des donnees manquantes
heart.mar.plot20 <- aggr(heart.mcar20, col=c('white','green4'),
                         numbers=TRUE, sortVars=TRUE,
                         labels=names(heart.mcar20), cex.axis=.9,
                         gap=3, ylab=c("Missing data","Pattern"))


##################################### KNN IMPUTATION ###################################

heart.knn20 <- kNN(heart.mcar20, k = 10, imp_var = FALSE)

err.knn20 <- mixError(heart.knn20, heart.mcar20, data.matrix(heart))


############################## Missforest IMPUTATION ###################################

heart.missforest20<-missForest(heart.mcar20, maxiter=10,
                               ntree = 200)

heart.missforest20$OOBerror




##################################### MICE IMPUTATION ##################################

# Imputation multiple
heart.mice20 <- mice(heart.mcar20, m = 1)

# Jeu de donnees impute
heart.mice.complet20 <- complete(heart.mice20)

# L'erreur
```

```
err.mice20 <- mixError(heart.mice.complet20, heart.mcar20, data.matrix(heart))


##################################### Amelia II #####################################

heart.amelia20 <- amelia(heart.mcar20 , m=5, noms = c(2,3,6,7,9,11,12,13,14))

err.amelia20 <- mixError(heart.amelia20$imputations$imp1, heart.mcar20,
                                                    data.matrix(heart))
```

Code 40% **données manquantes**

```
# Ratio de donnees manquantes
test.ratio <- 0.4

# Indices de l'echantillon test
IND <- which(!is.na(heart),arr.ind=TRUE)
ntest <- ceiling(nrow(heart) * test.ratio)
ind.test <- IND[sample(1:dim(IND)[1],ntest),]

# Creation des donnees manquantes
heart.test <- heart[ind.test]
heart.mcar40 <- heart
heart.mcar40[ind.test] <- NA

# Visualisation des donnees manquantes
heart.mar.plot40 <- aggr(heart.mcar40, col=c('white','green4'),
                         numbers=TRUE, sortVars=TRUE,
                         labels=names(heart.mcar40), cex.axis=.9,
                         gap=3, ylab=c("Missing data","Pattern"))


##################################### KNN IMPUTATION #####################################

heart.knn40 <- kNN(heart.mcar40, k = 10, imp_var = FALSE)

err.knn40 <- mixError(heart.knn40, heart.mcar40, data.matrix(heart))


################################ Missforest IMPUTATION #####################################

heart.missforest40<-missForest(heart.mcar40, maxiter=10,
                               ntree = 200)

heart.missforest40$OOBerror



##################################### MICE IMPUTATION #####################################
```

```
heart.mice40 <- mice(heart.mcar40, m = 1)  # imputation multiple

heart.mice.complet40 <- complete(heart.mice40)

err.mice40 <- mixError(heart.mice.complet40, heart.mcar40, data.matrix(heart))

################################### Amelia II ###################################

heart.amelia40 <- amelia(heart.mcar40 , m=5, noms = c(2,3,6,7,9,11,12,13,14))

err.amelia40 <- mixError(heart.amelia40$imputations$imp1, heart.mcar40,
                                                    data.matrix(heart))
```

9.1.2 MNAR

Code 10% **données manquantes**

```
heart.mnar10 <- ampute(data.matrix(heart), prop = 0.1, mech = "MNAR")
#prop = proportion
#mech = mechanisme

heart.mnar.plot10 <- aggr(heart.mnar10$amp, col=c('white','green4'),
                          numbers=TRUE, sortVars=TRUE,
                          labels=names(heart.mnar10$amp), cex.axis=.9,
                          gap=3, ylab=c("Missing data","Pattern"))


################################### KNN IMPUTATION ###################################

heart.knn10 <- kNN(heart.mnar10$amp, imp_var = FALSE)

err.knn10 <- mixError(heart.knn10, heart.mnar10$amp, data.matrix(heart))


############################## Missforest IMPUTATION ##############################

heart.missforest10<-missForest(heart.mnar10$amp, maxiter=10,
                               ntree = 200)
heart.missforest10$OOBerror


################################### MICE IMPUTATION ###################################

heart.mice10 <- mice(heart.mnar10$amp, m = 1)  # imputation multiple
heart.mice10$imp$chol
heart.mice.complet10 <- complete(heart.mice10)

err.mice10 <- mixError(heart.mice.complet10, heart.mnar10$amp,
                                            data.matrix(heart))
```

```
##################################### Amelia II ######################################

heart.amelia10 <- amelia(heart.mnar10$amp , m=1,
                                    noms = c(2,3,6,7,9,11,12,13,14))

err.amelia10 <- mixError(heart.amelia10$imputations$imp1, heart.mnar10$amp,
                                                  data.matrix(heart))
```

Code 20% données manquantes

```
heart.mnar20 <- ampute(data.matrix(heart), prop = 0.2, mech = "MNAR")


heart.mnar.plot20 <- aggr(heart.mnar20$amp, col=c('white','green4'),
                          numbers=TRUE, sortVars=TRUE,
                          labels=names(heart.mnar20$amp), cex.axis=.9,
                          gap=3, ylab=c("Missing data","Pattern"))


##################################### KNN IMPUTATION ##################################

heart.knn20 <- kNN(heart.mnar20$amp, imp_var = FALSE)

err.knn20 <- mixError(heart.knn20, heart.mnar20$amp, data.matrix(heart))


############################## Missforest IMPUTATION ##################################

heart.missforest20<-missForest(heart.mnar20$amp, maxiter=10,
                                    ntree = 200)

heart.missforest20$OOBerror



##################################### MICE IMPUTATION #################################

heart.mice20 <- mice(heart.mnar20$amp, m = 1)  # imputation multiple

heart.mice.complet20 <- complete(heart.mice20)


err.mice20 <- mixError(heart.mice.complet20, heart.mnar20$amp,
                                    data.matrix(heart))

##################################### Amelia II ######################################
```

```
heart.amelia20 <- amelia(heart.mnar20$amp , m=5,
                                        noms = c(2,3,6,7,9,11,12,13,14))

err.amelia20 <- mixError(heart.amelia20$imputations$imp1, heart.mnar20$amp,
                                                    data.matrix(heart))
```

Code 40% **données manquantes**

```
heart.mnar40 <- ampute(data.matrix(heart), prop = 0.4, mech = "MNAR")


heart.mnar.plot40 <- aggr(heart.mnar40$amp, col=c('white','green4'),
                          numbers=TRUE, sortVars=TRUE,
                          labels=names(heart.mnar40$amp), cex.axis=.9,
                          gap=3, ylab=c("Missing data","Pattern"))

########################################## KNN IMPUTATION ##################################

heart.knn40 <- kNN(heart.mnar40$amp, imp_var = FALSE)

err.knn40 <- mixError(heart.knn40, heart.mnar40$amp, data.matrix(heart))


############################### Missforest IMPUTATION ##################################

heart.missforest40<-missForest(heart.mnar40$amp, maxiter=20,
                               ntree = 300)

heart.missforest40$OOBerror


########################################## MICE IMPUTATION ##################################

heart.mice40 <- mice(heart.mnar40$amp, m = 1)  # imputation multiple

heart.mice.complet40 <- complete(heart.mice40)

err.mice40 <- mixError(heart.mice.complet40, heart.mnar40$amp,
                           data.matrix(heart))

########################################## Amelia II ##################################

heart.amelia40 <- amelia(heart.mnar40$amp , m=5,
                                        noms = c(2,3,6,7,9,11,12,13,14))

err.amelia40 <- mixError(heart.amelia40$imputations$imp1, heart.mnar40$amp,
                                                    data.matrix(heart))
```

9.1.3 MAR

Code 10% données manquantes

```
heart.mar10 <- ampute(data.matrix(heart), prop = 0.1, mech = "MAR")
#prop = proportion
#mech = mechanisme

heart.mar.plot10 <- aggr(heart.mar10$amp, col=c('white','green4'),
                         numbers=TRUE, sortVars=TRUE,
                         labels=names(heart.mar10$amp), cex.axis=.9,
                         gap=3, ylab=c("Missing data","Pattern"))


################################### KNN IMPUTATION ###################################

heart.knn10 <- kNN(heart.mar10$amp, imp_var = FALSE)

err.knn10 <- mixError(heart.knn10, heart.mar10$amp, data.matrix(heart))

################################ Missforest IMPUTATION ###############################

heart.missforest10 <- missForest(heart.mar10$amp, maxiter=10,
                                 ntree = 200)

heart.missforest10$OOBerror



################################### MICE IMPUTATION ##################################

heart.mice10 <- mice(heart.mar10$amp, m = 1)  # imputation multiple

heart.mice.complet10 <- complete(heart.mice10)

err.mice10 <- mixError(heart.mice.complet10,
                                          heart.mar10$amp, data.matrix(heart))

#################################### Amelia II #######################################

heart.amelia10 <- amelia(heart.mar10$amp , m=5,
                                          noms = c(2,3,6,7,9,11,12,13,14))

err.amelia10 <- mixError(heart.amelia10$imputations$imp1, heart.mar10$amp,
                                              data.matrix(heart))
```

Code 20% données manquantes

```
heart.mar20 <- ampute(data.matrix(heart), prop = 0.2, mech = "MAR")
```

```
heart.mar.plot20 <- aggr(heart.mar20$amp, col=c('white','green4'),
                         numbers=TRUE, sortVars=TRUE,
                         labels=names(heart.mar20$amp), cex.axis=.9,
                         gap=3, ylab=c("Missing data","Pattern"))


################################### KNN IMPUTATION ###################################

heart.knn20 <- kNN(heart.mar20$amp, imp_var = FALSE)

err.knn20 <- mixError(heart.knn20, heart.mar20$amp, data.matrix(heart))

############################## Missforest IMPUTATION ##################################

heart.missforest20<-missForest(heart.mar20$amp, maxiter=10,
                               ntree = 200)

heart.missforest20$OOBerror


################################### MICE IMPUTATION ##################################

heart.mice20 <- mice(heart.mar20$amp, m = 1)  # imputation multiple

heart.mice.complet20 <- complete(heart.mice20)

err.mice20 <- mixError(heart.mice.complet20,
                                          heart.mar20$amp, data.matrix(heart))


################################### Amelia II ########################################

heart.amelia20 <- amelia(heart.mar20$amp , m=1,
                                          noms = c(2,3,6,7,9,11,12,13,14))

err.amelia20 <- mixError(heart.amelia20$imputations$imp1, heart.mar20$amp,
                                          data.matrix(heart))
```

Code 40% **données manquantes**

```
heart.mar40 <- ampute(data.matrix(heart), prop = 0.4, mech = "MAR")

heart.mar.plot40 <- aggr(heart.mar40$amp, col=c('white','green4'),
                         numbers=TRUE, sortVars=TRUE,
                         labels=names(heart.mar40$amp), cex.axis=.9,
                         gap=3, ylab=c("Missing data","Pattern"))

################################### KNN IMPUTATION ###################################

heart.knn40 <- kNN(heart.mar40$amp, imp_var = FALSE)
```

```
err.knn40 <- mixError(heart.knn40, heart.mar40$amp, data.matrix(heart))


############################## Missforest IMPUTATION ###################################

heart.missforest40<-missForest(heart.mar40$amp, maxiter=10,
                               ntree = 200)

heart.missforest40$OOBerror



################################### MICE IMPUTATION ###################################

heart.mice40 <- mice(heart.mar40$amp, m = 1)  # imputation multiple

heart.mice.complet40 <- complete(heart.mice40)

err.mice40 <- mixError(heart.mice.complet40, heart.mar40$amp,
                                                data.matrix(heart))

#################################### Amelia II ########################################

heart.amelia40 <- amelia(heart.mar40$amp , m=5,
                                           noms = c(2,3,6,7,9,11,12,13,14))

err.amelia40 <- mixError(heart.amelia40$imputations$imp1, heart.mar40$amp,
                                           data.matrix(heart))
```