S6_1_Donnees_manquantes

Table of contents

1 Introduction aux données manquantes

1. Considérer le problème = **décrire les données manquantes** : combien ? où ? pourquoi ?
2. Gestion des données manquantes
   - **Virer les sujets avec données manquantes** (listwise deletion) = `na.rm=TRUE` : bof notamment si rajout de variables dans un modèle -> perte de sujets car modèle ajusté se fait sur les sujets avec données complètes
   - **Imputation des données manquantes** (remplir les valeurs manquantes avec des estimations)
     - **Imputation simple** : mode (valeur la plus fréquente) pour les variables catégorielles, moyenne/médiane pour les variables continues (Library `Hmisc` : fonction `impute()` (imputation simple))
     - **Imputation multiple** : créer plusieurs jeux de données imputées, analyser chaque jeu de données, combiner les résultats (Library `mice` : fonction `mice()` (imputation multiple))

2 Types de données manquantes

2.A MCAR : Missing Completely At Random

Les données manquantes sont indépendantes des valeurs observées et non observées. Exemple : perte de questionnaire.

= Données manquantes **complètement au hasard**

2.B MAR : Missing At Random

Les données manquantes **dépendent des valeurs observées** mais **pas des valeurs non observées**.

Exemple :

- les personnes âgées sont moins susceptibles de remplir un questionnaire en ligne.
- les patients très musulmans répondent moins sur leur consommation d'alcool, mais on connaît leur âge et sexe.
  - donc religion TRUE –> + de données manquantes et moins d'alcool
  - religion FALSE –> peu de données manquantes, + d'alcool
  - si on a religion dans le questionnaire : c'est ok ! sinon on sait pas pourquoi alcool est manquant (biais dans la donnée manquante)

Une fois qu'on enlève l'effet de toutes les variables mesurées susceptibles d'influencer à la fois la réponse d'une variable mesurée et à la fois la probabilité que cette variable soit manquante, ce qui reste dans la valeur manquante de la donnée : c'est du bruit

Autrement dit : une fois ajusté sur les variables mesurées pertinentes, le caractère manquant d'une observation ne dépend plus de sa valeur non observée.

> **!** Important
>
> **MAR** : le fait qu'une donnée soit manquante dépend uniquement des autres variables observées dans le jeu de données, pas de sa propre valeur non observée.
> Une fois ces variables observées prises en compte, le caractère manquant devient aléatoire.

2.C MNAR : Missing Not At Random

Les données manquantes **dépendent des valeurs non observées**.

Exemple : les patients avec une dépression sévère sont moins susceptibles de remplir un questionnaire sur la dépression.

= Données manquantes **non au hasard**

Pour faire la différence entre MCAR et MNAR :

- Les tests statistiques ne marchent pas terrible
- Souvent, on fait des hypothèses basées sur la connaissance du domaine

3 Imputation multiple

Surtout si bcp de données ou manquantes ou si données manquantes sur données importantes

3.A Principe de l'imputation multiple

Dans un modèle :

$Y = a + bX_1 + cX_2 + \epsilon$

On s'intéresse au paramètre b.

Jeu de données :

ID	Y	X1	X2
1	2.3	1.2	NA
2	3.1	NA	0.5
3	NA	0.8	1.1
4	4.0	1.5	0.9
5	2.8	1.0	0.7

Pour faire une imputation multiple :

3.A.1 1ère étape : tirage au sort

- Tirer au sort la valeur manquante de Y par une autre valeur de Y (`df[3, "Y"]` est manquant, on le remplace par `df[5, "Y"]` = 2.8)
- Idem pour X1 : `df[2, "X1"]` est manquant, on le remplace par `df[1, "X1"]` = 1.2
- Idem pour X2 : `df[1, "X2"]` est manquant, on le remplace par `df[5, "X2"]` = 0.7

On obtient un jeu de données **INTERMÉDIAIRE** qui permet de réaliser une régression complète

3.A.2 2ème étape : régressions

3 régressions sur le jeu de données intermédiaire :

- $Y[3] = a + bX_1[3] + cX_2[3] + \epsilon$ permet d'obtenir la valeur estimée de Y[3]
- $X_1[2] = a' + b'Y[2] + c'X_2[2] + \epsilon'$ permet d'obtenir la valeur estimée de X1[2]
- $X_2[1] = a'' + b''Y[1] + c''X_1[1] + \epsilon''$ permet d'obtenir la valeur estimée de X2[1]

Ainsi on obtient un jeu premier jeu données **IMPUTÉ** mais pas encore **convergent**

Convergent : plus on impute, plus les jeux de données se ressemblent

3.A.3 3ème étape : itérations

On recommence les étapes 1 et 2 plusieurs fois (10 à 20 fois) pour obtenir un jeu de données **IMPUTÉ CONVERGENT**

Quand le jeu de données a convergé, il est parfaitement bien imputé car toutes les données manquantes sont construites à partir de toutes les données disponibles sur les autres variables.

3.A.4 4ème étape : répétition

- Il faut **REFAIRE LE TIRAGE AU SORT** de départ pour obtenir un nouveau jeu de données intermédiaire
- Puis refaire les étapes 2 et 3 pour obtenir un **NOUVEAU JEU DE DONNÉES IMPUTÉ CONVERGENT**

On a donc un **DEUXIÈME JEU DE DONNÉES IMPUTÉ CONVERGENT**

Puis répétition N fois (en général 5 à 10 fois) pour obtenir N jeux de données imputés convergents.

3.A.5 5ème étape : analyse

On réalise l'analyse statistique (régression linéaire, logistique, etc.) sur chaque jeu de données imputé convergent.

jeu ' : $Y^{(')} = a^{(')} + b^{(')}X_1 + c^{(')}X_2 + \epsilon^{(')}$

jeu 1 : $Y^{(1)} = a^{(1)} + b^{(1)}X_1 + c^{(1)}X_2 + \epsilon^{(1)}$

jeu 20 : $Y^{(20)} = a^{(20)} + b^{(20)}X_1 + c^{(20)}X_2 + \epsilon^{(20)}$

On obtient donc

- 20 estimations différentes de b.
- avec 20 variances de b différentes. (variance inter-modèle)
- et une variance de b1 à b20 = variance intra-modèle

Prendre :

- La moyenne des 20 estimations de b comme estimation finale de b.
- − variance inter-modèle + variance intra-modèle pour obtenir la variance finale de b.

b moyen = estimation non biaisée de b

somme des variances intra + inter-modèle = variance non biaisée de b

3.B Gérer les analyses après imputation multiple

20 jeu de données imputés !