Traitement des données manquantes

Julien JACQUES

Motivations

```
library(missMDA)
data(geno)
summary(geno)
```

```
##       ACOR               AORE               ASAL               CALB
##  Min.   :-1.95016   Min.   :-1.9533   Min.   :-1.75478   Min.   :-0.44588
##  1st Qu.:-0.87034   1st Qu.:-0.4141   1st Qu.:-0.38103   1st Qu.:-0.13087
##  Median :-0.01228   Median : 0.1299   Median : 0.04972   Median :-0.04788
##  Mean   : 0.00000   Mean   : 0.0000   Mean   :-0.02240   Mean   :-0.01161
##  3rd Qu.: 0.85716   3rd Qu.: 0.5368   3rd Qu.: 0.49009   3rd Qu.: 0.15438
##  Max.   : 2.17284   Max.   : 1.4987   Max.   : 0.62447   Max.   : 0.55512
##                                       NA's   :1          NA's   :3
##       CBAD               CCOR               CLER               CSE1
##  Min.   :-0.64487   Min.   :-0.50434   Min.   :-1.321063   Min.   :-0.56372
##  1st Qu.:-0.32412   1st Qu.:-0.18634   1st Qu.:-0.462813   1st Qu.:-0.41559
##  Median : 0.06788   Median : 0.04916   Median : 0.196187   Median : 0.13453
##  Mean   : 0.02236   Mean   : 0.00000   Mean   :-0.003196   Mean   : 0.03365
##  3rd Qu.: 0.26462   3rd Qu.: 0.19153   3rd Qu.: 0.502062   3rd Qu.: 0.40241
##  Max.   : 0.71637   Max.   : 0.35791   Max.   : 0.608437   Max.   : 0.80053
##  NA's   :3                             NA's   :1           NA's   :1
##       CSE2               CTO1
##  Min.   :-1.09084   Min.   :-0.530281
##  1st Qu.:-0.49247   1st Qu.:-0.245531
##  Median : 0.09566   Median : 0.071969
##  Mean   :-0.02518   Mean   : 0.006719
##  3rd Qu.: 0.52003   3rd Qu.: 0.239094
##  Max.   : 1.03366   Max.   : 0.359969
##  NA's   :1          NA's   :1
```

Les différentes types de données manquantes

- MCAR (Missing Completely At Random) : l'absence de données est complétement aléatoire, cela ne dépend pas des autres variables observées.
- MAR (Missing At Random) : l'absence de données est aléatoire, mais les autres variables peuvent permettre de prédire la donnée manquante
- MNAR (Missing Not At Random) : l'absence de données dépend de la valeur elle même

Références :
Statistical Analysis with Missing Data, 3rd Edition, Roderick J. A. Little, Donald B. Rubin, April 2019
https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-idm.pdf

Comment gérer les données manquantes (MCAR, MAR)

- les ignorer : supprimer les obserations ayant des données manquantes, si leur proportion est faible
- les imputer : on chercher à remplacer la valeur manquante par une valeur qui soit la plus proche de ce qui aurait dû être observé
- utiliser une méthode qui permette de les traiter comme telle : le processus d'estimation tien compte de ces données manquantes

Méthode d'imputation simple, MCAR

La façon la plus classique de traiter les données manquantes, est de les **imputer** au préalable de l'analyse statistique que l'on veut mener.

Imputer les données manquantes MCAR, dans le cas de **variables continues**.

MCAR signifie que les autres variables ne servent à rien. On va imputer en fonction de la distribution de la variable elle même :

- par la moyenne :

```
geno$ASAL[is.na(geno$ASAL)]=mean(geno$ASAL,na.rm=T)
```

- par la médiane (moins sensibles données atypiques...) :

```
geno$CALB[is.na(geno$CALB)]=median(geno$CALB,na.rm=T)
```

Méthode d'imputation simple, MCAR

Cas de **variables catégorielles** :

```
data(vnf)
summary(vnf)
```

```
##    Q7.1          Q7.2          Q7.4          Q8.1          Q8.2
##  1   :776    1   :424    1   :909    1   :884    1   :894
##  2   :305    2   :314    2   :217    2   :323    2   :310
##  3   :102    3   :407    3   : 66    NA's: 25    NA's: 28
##  NA's: 49    NA's: 87    NA's: 40
##   Q29.2         Q29.3         Q30.1         Q30.2         Q30.3
##  1   :685    1   :259    1   :566    1   :514    1   :134
##  2   :475    2   :635    2   :380    2   :419    2   :558
##  NA's: 72    3   :192    NA's:286    NA's:299    3   :147
##              NA's:146                            NA's:393
```

Méthode d'imputation simple, MCAR

Imputer les données manquantes MCAR, dans le cas de **variables catégorielles** :

- par le mode :

```
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
vnf$Q7.1[is.na(vnf$Q7.1)]=getmode(vnf$Q7.1)
```

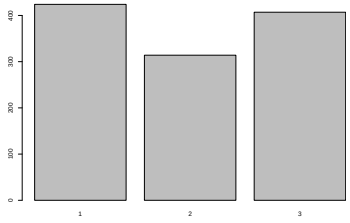
Mais attention, cela peut déséquilibrer les proportions initiales en cas de grande proportion de données manquantes.

Méthode d'imputation simple, MCAR

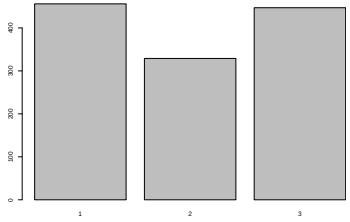
Imputer les données manquantes MCAR, dans le cas de **variables catégorielles** :

- en tirant aléatoirement suivant la distribution de la variable :

```
plot(vnf$Q7.2)
```



```
vnf$Q7.2[is.na(vnf$Q7.2)]=sample(levels(vnf$Q7.2),
    size=sum(is.na(vnf$Q7.2)),replace=T,
    prob = table(vnf$Q7.2))
plot(vnf$Q7.2)
```



Méthode d'imputation simple, MAR

Dans le cas MAR, les autres variables peuvent nous aider à prédire la valeur manquantes.

Dans ce cas, on va chercher à construire un **modèle prédictif** qui permette de prédire la variable dont la valeur est manquante.

Des packages sont spéficiquement dédiés à cela :

- basé sur des reconstruction par analyse factorielle :

```
library(missMDA)
```

- basé sur un processus d'*équations chainées*

```
library(mice)
```

Méthode d'imputation multiple

La valeur imputée peut avoir un impact sur l'analyse effectuée.

Idéalement, il faudrait étudier la sensibilité de l'analyse à cette imputation.

Pour cela, des techniques sont basées sur des **imputations multiples** :

- on propose plusieurs valeurs d'imputation
- pour chaque valeur, on estime le modèle
- et on construit un méta-modèle qui mixe ces différents modèles

Cela permet en outre d'évaluer l'incertitude due à l'imputation.

Méthode avancée

Une alternative à l'imputation est d'estimer le modèle en tenant compte de la présence de données manquantes.

Des algorithmes d'optimisation de type EM sont adaptées à maximiser une vraisemblance en présence de données manquantes.

Ils fonctionnent en itérants jusqu'à convergence les deux étapes suivantes, en partant d'une initisalition aléatoire des paramètres du modèle :

- on impute les données manquantes en utilisant le modèle courant
- on met à jour les paramètres du modèle avec les nouvelles valeurs imputées

Le paquet suivant permet cela dans le cas de la régression linéaire et logistique :

```
library(misaem)
```