"Quaderni di Ricerca in Didattica (Mathematics)", n. …, 2010
G.R.I.M. (Department of Mathematics, University of Palermo, Italy)
A.S.I. 5 Proceedings 5-7- November 2010

Imputation multiple de données manquantes par l'Analyse Statistique Implicative

Pablo Gregori[1]*, Josep V. Felip-Bardoll[2] et Régis Gras[3]
[1]Departament de Matemàtiques
Institut Universitari de Matemàtiques i Aplicacions de Castelló IMAC
Universitat Jaume I de Castellón
E-12071 Castelló de la Plana (Spain)

[2]Centre d'Informàtica,
Conselleria de Sanitat,
Av. Campanar, 21
E-46009 València (Spain)

[3]Ecole Polytechnique de l'Université de Nantes, Equipe Connaissance et Décision,
Laboratoire d'Informatique de Nantes-Atlantique (LINA), UMR 6241,
La Chantrerie – BP 50609, 44306 Nantes cedex

E-mail: gregori@mat.uji.es, felip_jvibar@gva.es, regisgra@club-internet.fr

Résumé. La plupart des études de recherche appliquée nécessitent des traitements sur des tableaux de données qui présentent fréquemment des données manquantes. Or la majorité des algorithmes statistiques ne travaillent que sur des tableaux de données complets. Donc chaque individu (file) se présentant avec une ou plusieurs lacunes doit être ignoré. L'imputation est une procédure qui consiste à mettre des valeurs dans les cases vides du tableau pour tirer parti des individus incomplets. Gras (2009) propose une méthode originale d'imputation basée sur l'Analyse Statistique Implicative (A.S.I.). Dans cette contribution, on analyse l'applicabilité de cette méthode et on propose l'utilisation de l'imputation multiple pour obtenir des estimations d'intensités d'implication d'un tableau de données obtenues d'une enquête sociologique.

Abstract. Most of applied research studies are based on datasheets presenting a fraction of missing data. Statistical algorithms can only be applied to complete datasheets, then every case (row) with one or more gaps must be fully discarded. Imputation of missing data is a step that allows one for taking profit of the information provided by those incomplete cases. Gras (2009) proposes an original data imputation method based on Statistical Implicative

* First author acknowledges the financial support provided by Bancaja—UJI through project P1·1B2008-27

Analysis. In this contribution we analyze the applicability of this method and we propose the use multiple imputation, in order to provide estimations on the implication intensities of a data table issued from a sociological survey.

1 Introduction

Les tableaux de données couramment utilisés dans l'analyse statistique présentent souvent des données manquantes. Par exemple, il est fréquent que les personnes qui ont volontairement répondu à un sondage d'opinion refusent de répondre à certaines questions qu'ils considèrent trop personnelles, voire intimes.

Les techniques statistiques partent des données pour apporter de l'information, et ne sont généralement pas propres à traiter des ensembles de données dans lesquels certaines d'entre elles sont inconnues. Les deux formes, radicalement différentes, pour contourner ou résoudre le problème sont les suivantes :

1. Elimination: réduire l'échantillon aux individus «complets», parmi lesquels on ne trouve aucune donnée manquante.
2. Imputation : attribuer des valeurs aux données manquantes selon un critère rationnel.

Retirer les individus qui présentent des données manquantes, s'ils représentent un faible pourcentage de l'échantillon (5% ou moins), peut paraître raisonnable. Sinon, cette procédure introduit un biais dans les conclusions extraites. En outre, l'affectation de données créées, variable par variable (par exemple en utilisant les valeurs moyennes), conduit à une réduction systématique de la dispersion de chaque variable et risque de briser, sans doute, d'éventuelles relations multidimensionnelles sous-jacentes. Cependant, même si cela se fait dans le respect des relations entre les variables, les différences provenant des méthodes d'imputation, même non aléatoires, ne permettent pas l'expression d'inférences sur les résultats des analyses associées. Ensuite, quelle que soit la méthode, on refait l'analyse avec la nouvelle table de données et on obtient des conclusions, de la même façon que si le tableau original de données incomplètes n'avait pas existé.

Une piste pour réussir à capter la variabilité causée par l'absence de données consiste en l'utilisation intensive d'une « bonne » affectation tenant compte de la nature des données et de l'analyse de la variabilité produite par cette série d'imputations. On trouvera deux excellentes études sur l'analyse statistique avec données manquantes dans Schafer (1979) et Little & Rubin (1987). On conçoit bien que l'ordinateur jouera un rôle essentiel dans le type de traitement qui consiste

à simuler et mettre en œuvre de façon intensive des algorithmes statistiques. L'environnement informatique de ce travail a été le logiciel R (R, 2008).

Dans cette contribution on utilisera les outils de l'Analyse Statistique Implicative (ASI) dans deux buts : d'abord, pour élaborer la méthode d'imputation (Gras, 2009), ensuite pour illustrer les possibilités inférentielles de l'imputation multiple. Rappelons que l'ASI est une méthodologie développée et utilisée dans diverses disciplines, initialement dans Gras (1979), puis Gras et al. (1996), jusqu'à Gras et al. (2008), Gras et al. (2009) et Orús et al. (2009). C'est pour cette raison qu'on a utilisé le logiciel CHIC 4.1 (Couturier & Ag Almouloud, 2009; Ratsimba-Rajohn, 2009), le seul logiciel conçu spécifiquement pour ce type d'analyse et d'une confortable interface graphique pour l'utilisateur.

La section 2 présente le tableau de données original qui servira à illustrer nos propositions. Dans la section 3, on rappelle des méthodes d'imputation les plus courantes, et on continue en suivant la proposition de Gras (2009) dans la section 4. Ensuite, on compare les méthodes d'imputation avec la partie complète du tableau disponible dans la section 5. Dans la section 6 on propose un modèle d'imputation multiple : celui défini par l'application intensive d'une affectation aléatoire basée sur celle de Gras (2009). Enfin, on analyse les différentes possibilités de calcul sur les données disponibles dans la section 7, et on dégage des conclusions (section 8).

2 Tableau de données

Le tableau de données que l'on traite ici est un sous-ensemble du tableau obtenu dans une enquête sociologique (FUNDESTAP, 2010) qui s'est déroulée dans le village d'Alcublas, dans la province de Valence (Espagne). On a recueilli des renseignements auprès d'un échantillon de 146 personnes (représentants des 829 habitants recensés à 2007) selon 103 variables portant sur la satisfaction relativement aux services de gestion, d'installations et d'infrastructures du village. Pour ce travail, on n'a retenu que les 37 variables « sociales ». Le tableau de données original est construit à partir d'une échelle de Likert de cinq niveaux de satisfaction ; on offre la possibilité de répondre « Ne connais pas » et « Pas de réponse ».

Afin d'utiliser l'ASI dans sa version binaire on a codé « 1 » les réponses des niveaux 4 et 5 sur l'échelle de satisfaction, et « 0 » le reste. Ainsi, les variables sont interprétées comme la présence d'une satisfaction positive des aspects évalués. On a pris les réponses « Ne connais pas » et « Pas de réponse » comme des réponses manquantes, car elles n'appartiennent pas à l'échelle de satisfaction d'origine.

3 Les méthodes d'imputation

Dans la théorie de l'analyse statistique avec des données manquantes, on peut considérer trois hypothèses différentes sur l'origine du mécanisme sous-jacent à la non réponse dans les tableaux de donnés : MCAR (missingness completely at ramdom), où le mécanisme de non réponse est indépendant des valeurs observées : MAR (missingness at ramdom), où le mécanisme de non réponse dépend des valeurs observées ; et MNAR (missingness not at ramdom), où le mécanisme de non réponse dépend aussi des valeurs non observées. C'est la première hypothèse qui simplifie au mieux les analyses et c'est celle que l'on va adopter dans cette approche.

Les deux méthodes les plus courantes dans les recherches sur l'imputation sont les suivantes:

- Imputation par la moyenne ou le mode: on attribue la valeur moyenne ou la plus observée de la variable à chacune des données manquantes de la même variable. Cette approche conduit à sous-estimer la dispersion des variables et à briser la structure de la relation entre elles.

- Imputation utilisant la régression: les variables avec données manquantes sont traitées comme des variables dépendantes, et une analyse de régression (linéaire ou logistique, selon le cas) sur le reste de variables permet l'affectation. Dans ce cas, on réussit à maintenir la structure de la relation entre les variables, bien qu'existe le risque que les données ne suivent pas la régression choisie par l'utilisateur.

On montre dans l'annexe 3 l'implémentation d'une fonction dans le langage R de ces méthodes d'imputation.

Une autre méthode pour traiter les données manquantes est dénommée « méthode de la variable indicatrice » (Jones, 1996). Dans ce cas, on crée une nouvelle variable binaire associée à chaque variable présentant des données manquantes, de sorte que, pour chacune des variables ayant une ou plus d'une donnée manquantes, va apparaître une nouvelle variable à coté. Sur un sujet n'ayant pas de donnée manquante, la valeur de la variable reste inchangée, tandis que la variable nouvelle (indicatrice) associée, si elle existe, est mise à «0».D'autre part, sur un sujet présentant la donnée manquante, la valeur inconnue est imposée à « 0 » (par exemple, si numérique) ou à une catégorie « extra » (si qualitative), et la variable indicatrice prend la valeur « 1 ». Finalement, les nouvelles variables (indicatrices) ainsi que les affectations sur les variables originales deviennent parties intégrantes du tableau de données, et les analyses continuent.

Ces méthodes, comme beaucoup d'autres qui ne sont pas mentionnés ici, sont déterministes, car il n'y a qu'une seule issue possible après l'imputation. En revanche, la méthode de l'imputation multiple (Rubin, 1987) est l'un des plus attrayantes pour gérer le manque de données. Elle utilise des affectations aléatoires. L'idée de base peut être résumée selon les étapes suivantes:

1. Affecter des valeurs aux données manquantes en utilisant un modèle aléatoire « approprié ».
2. Répéter M fois l'étape 1, aboutissant à M tableaux de données complets.
3. Effectuer l'analyse désirée sur l'ensemble des tableaux de M.

A la fin de ces étapes, les résultats de l'analyse de l'étape 3 peuvent être « moyennés » si l'on veut estimer la valeur et mesurer la stabilité en calculant l'erreur quadratique moyenne, etc. Rubin (1987) a montré qu'un faible nombre d'affectations (M = 3, 4, 5) conduit à de bonnes estimations. Dans notre cas, on ne va pas avoir un candidat objectivement approprié pour le modèle aléatoire utilisé à l'étape 1, si bien qu'on va le définir selon la philosophie de l'ASI. En conséquence, on va simuler un grand nombre de fois, puisque l'outil de calcul va le permettre dans un court délai.

4 Imputation basée sur l'ASI

Cette section décrit brièvement la méthode proposée par Gras (2009) pour affecter des valeurs aux données manquantes. On utilise pour cela une mesure de proximité qui prend en compte toutes les variables, c'est-à-dire que l'on calcule la distance entre l'individu avec la donnée manquante, par exemple x, et les autres individus dans l'échantillon, dont on connaît les données sur cette variable. Ainsi, en choisissant le sujet le plus proche, soit y_0, la donnée manquante « hérite » la valeur de cet individu.

Soit un tableau de données qui croise un ensemble de variables V sur un ensemble de sujets E, de façon que le sujet x montre une absence de réponse sur la variable i. Alors, pour « remplir » son absence de réponse on adopte la méthodologie aux étapes suivantes, révélant le cheminement épistémologique de la procédure : pour chaque autre sujet du tableau, y, on calcule :

V_{xy} : le sous-ensemble de variables de V pour lesquelles les sujets x et y prennent en commun des valeurs. Soient v_{xy} et v leurs cardinaux respectifs.

Marges des sujets *x* et *y*, respectivement $x_{\cdot} := \sum_{j \in V_{xy}} x(j)$ et $y_{\cdot} := \sum_{j \in V_{xy}} y(j)$.

Profils des sujets *x* et *y*, respectivement $\left\{ \frac{x(j)}{x_{\cdot}} \right\}_{j \in V_{xy}}$ et $\left\{ \frac{y(j)}{y_{\cdot}} \right\}_{j \in V_{xy}}$.

Marge de chaque variable $j \in V_{xy}$, respectivement $n_j := \sum_{z \in E_j} z(j)$, où E_j est le sous-ensemble des sujets pour lesquels la variable *j* n'est pas absente.

Marge totale $N := \sum_{j \in V_{xy}} n_j$.

Donc on prend les sujets comme distribués le long des variables connues, et on calcule leur écart par une distance de type χ^2, soit :

$$d(x,y) := \left[\sum_{j \in V_{xy}} \frac{\left(\frac{x(j)}{x_{\cdot}} - \frac{y(j)}{y_{\cdot}} \right)^2}{n_j / N} \right]^{1/2}$$

distance que l'on corrige par un facteur attribuant plus d'importance aux individus *y* qui ont plus de variables en commun avec *x*, de la forme $\delta(x,y) := \left(1 - \frac{v_{xy}}{v}\right) d(x,y)$. Toujours pour une raison visant le respect de la sémantique du problème, (Gras, 2009, section 4) propose de plus une variante : intégrer la relation implicative entre les variables dans la procédure d'affectation des données manquantes. Plus précisément:

1. On prend comme imputation initiale celle calculée à partir des proximités avec les autres sujets du tableau.

2. On choisit un seuil pour en considérer les implications importantes sous-jacentes au tableau.

3. Pour chaque donnée manquante (disons, relativement à la variable *i*):

    a. On réunit dans une classe de variables *C* toutes les variables *j* de sorte que la règle $j \rightarrow i$ ait une intensité d'implication au-dessus du seuil.

    b. On réunit dans une classe *C'* les variables *k* telles que l'intensité de l'implication de la règle $i \rightarrow k$ dépasse aussi le seuil.

4. Parmi les deux classes de variables *C* et *C'*, on choisit la classe qui a une plus grande cohésion.

5. Parmi les variables de la classe choisie, on tire celle dont la règle implicative ($j \rightarrow i$ ou $i \rightarrow k$, selon le cas) a une valeur d'intensité plus élevée.

6. L'imputation initiale *x(i)* se modifie comme suit:

    a. Si *C* et j_0 ont été la classe et la variable choisies, on prend alors $x(i) := \max\{y_0(i), x(j_0)\}$ comme nouvelle imputation. Cela signifie que l'imputation initiale, $y_0(i)$ (la valeur de l'individu plus ressemblant à celui de la donnée manquante), est modifiée par $x(j_0)$, si celui-ci améliore la qualité de l'implication particulière $j_0 \rightarrow i$.

    b. De même, dans le but de préserver la relation implicative la plus importante qui concerne la variable à compléter, si *C'* et k_0 sont la classe et la variable choisies, on prend $x(i) := \min\{y_0(i), x(k_0)\}$ comme nouvelle implication.

À l'implémentation informatique de l'algorithme décrit dans Gras (2009), et développé dans l'annexe 5, on a introduit la nuance, rigoureusement non négligeable, suivante : ne pas utiliser les imputations des données manquantes faites auparavant dans les calculs pour les imputations à faire dorénavant. Cela garantit l'indépendance du résultat final de l'ordre d'imputation choisi.

5 Comparaison des méthodes d'imputation

Afin de comparer les méthodes d'imputation par le mode (MODE), la régression logistique (REG) et celle basée sur l'ASI (ASI), on prélève, du tableau de données disponible (146x37), le sous-ensemble d'individus complets (23x37).

Avec ce tableau d'individus complets on va comparer l'**effectivité** des trois méthodes mentionnées sous l'hypothèse MCAR. Donc on simule une « non réponse » complètement au hasard dans le tableau, on pratique les imputations sur ces données manquantes au moyen de chaque méthode, on enregistre le pourcentage de succès apportés par chaque méthode (nombre de valeurs imputées qui coïncident avec les valeurs effacées sur le nombre de valeurs effacées), et on répète l'opération 1000 fois afin de calculer les taux d'effectivité. Pour obtenir des résultats plus complets, on choisit de créer trois scénarios differents : un avec un 5% de données manquantes, un autre avec un 10%, et un dernier avec un 20%. La figure 1 compare la distribution des taux d'effectivité donnés par les trois méthodes dans chacun des trois scénarios.

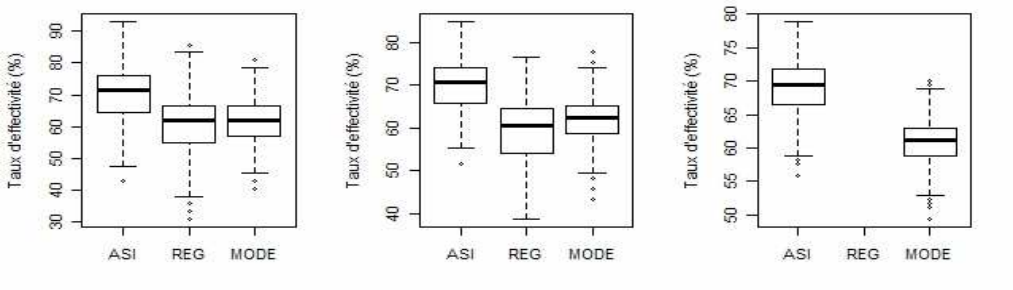

Figure 1. Taux d'effectivité des méthodes d'imputation comparées après 1000 simulations avec un 5% (gauche), 10% (centre) et 20% (droit) de données manquantes.

On s'aperçoit à droite (20% de données manquantes) que la méthode REG ne réussit pas à donner une imputation satisfaisante sur les données manquantes permettant de calculer le taux d'effectivité. Au centre, le diagramme de Tuckey de la méthode REG n'a pu être établi qu'à partir des 60 (sur 1000) imputations satisfaisantes. La méthode d'imputation qui utilise la régression présente donc une faiblesse importante quand le nombre de valeurs manquantes est grand, peut-être à cause de la taille du tableau, car la simulation de MCAR produit très souvent des tableaux où le nombre d'individus complets se réduit tellement, que la fonction de régression n'est pas calculable. La méthode utilisant le mode reste aussi en deçà de l'ASI, en particulier lorsque le nombre d'absences croît. **La méthode basée sur l'ASI se montre ainsi la plus efficace** des trois méthodes : taux d'effectivité le meilleur et dispersion la moins grande

L'annexe 6 contient le petit code-programme exécuté pour procéder à la simulation, l'obtention des taux d'effectivité et les graphes de la figure 1. Mais comme on a gardé les données originales pour des raisons de droits de protection, on peut utiliser ce code en remplaçant le nom du fichier par un autre avec les données d'intérêt du lecteur.

6 Imputation multiple basée sur l'ASI

Afin d'obtenir une méthode d'imputation multiple basée sur l'ASI, on prendra la méthode d'imputation de Gras (2009) à partir de laquelle on créera une imputation aléatoire. Pour chaque donnée manquante :

1. On calcule les distances entre l'individu de la donnée manquante et les autres individus du tableau de données ($\delta(x, y)$).

2. A chaque individu *y* on lui attribue un poids en fonction de sa distance à *x*.

3. On échantillonne dans la population des individus $y \in E$ avec des probabilités proportionnelles à leur poids, c'est-à-dire, avec probabilité $\frac{Poids(y)}{\sum_{y \in E} Poids(y)}$.

4. Finalement, on propose comme affectation à la donnée manquante la valeur relative à l'individu échantillonné dans l'étape 3.

Le caractère aléatoire de ce modèle d'imputation réside dans le point 3, puisque les individus choisis pour l'imputation sont échantillonnés à chaque fois avec des probabilités proportionnelles à leur proximité aux individus dont leurs données manquantes vont être affectées. C'est pour cela qu'on pourra appliquer l'imputation multiple de Rubin (1987) et essayer de saisir la variabilité causée par l'absence de certaines données, dans le tableau, comme dans n'importe quelle analyse statistique. Dans la section 6 on montre, en particulier, comment l'imputation multiple apporte une sorte d'inférence sur les résultats des intensités d'implication quand ils ont été appliqués sur un tableau incomplet.

On propose le choix des poids suivant : à partir d'une fonction $f : [0,\infty) \to [0,\infty)$ non croissante, pas nécessairement continue, on définit:

$$Poids(y) := \begin{cases} 0 & , \text{si } y(i) \text{ est manquante} \\ f(\delta(x,y)) & , \text{sinon} \end{cases}$$

Dans l'algorithme programmé (dans l'annexe 5) on a pris, en particulier, la fonction

$$f(t) := \begin{cases} 2 & , t = 0 \\ e^{-2t} & , t > 0 \end{cases}$$

De cette sorte les données manquantes seront plutôt affectées par des valeurs des individus les plus proches, mais pas dans tous les cas, et on aura ainsi saisi une variabilité liée à l'ignorance des valeurs réelles.

7 Exemple de matrice implicative avec des données manquantes et imputation multiple

Les calculs exécutés dans la plupart des analyses statistiques nécessitent un tableau complet. Si on prend la décision d'attribuer des valeurs aux donnés manquantes, une attribution unique aboutit à un unique résultat de l'analyse choisie, ce qui risque de conduire à une information plutôt réduite, à partir du manque des

données. En revanche, une application intensive d'attributions selon un modèle convenable devient un échantillon particulier de résultats de l'analyse, qui informe sur les vraisemblances des résultats eux-mêmes. Donc il apporte une sorte d'« intervalle de confiance » sur le véritable résultat inconnu de l'analyse.

On illustre cette idée par le calcul de la matrice d'intensités d'implication avec le tableau utilisé dans la section 5. Après une élimination aléatoire du 20% des entrées du tableau de 23x37, l'imputation multiple (M = 1000) basée sur l'ASI, on n'a conservé, pour des raisons de place, que les 4 variables ayant la plus grande dispersion des valeurs d'intensités d'implication et le plus grand nombre de données manquantes (7 pour chacune). Les résultats de la matrice implicative apparaissent dans la figure 2.

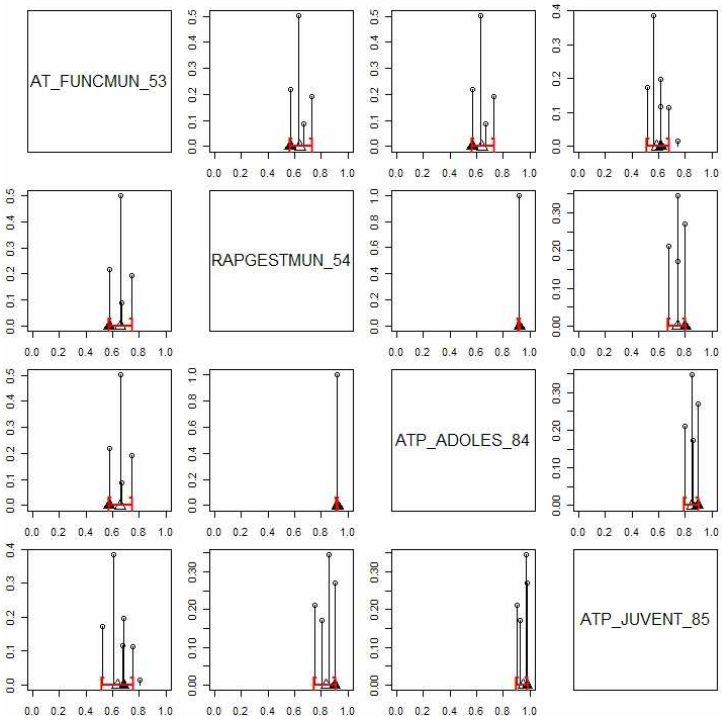

Figure 2. Distribution d'échantillonnage de l'intensité de chaque implication (file → colonne) dans le tableau de données utilisé (voir section 5) sous l'imputation multiple basée sur l'ASI, exprimée par les frequences relatives de chaque valeur observée.

On a dessiné un intervalle du 95% des valeurs de l'échantillon, de la même façon qu'un intervalle de confiance. Par exemple, le graphe de la 1ère file et 3ème colonne représente la distribution des 1000 valeurs de l'intensité de l'implication AT_FUNCMUN_53 --> ATP_ADOLES_84. On a tracé des segments verticaux pour chaque valeur observée de hauteur égale à sa fréquence relative. De plus, sur l'axe horizontal on a indiqué la valeur de l'intensité d'implication pour le tableau

complet (triangle noir), la moyenne des 1000 valeurs obtenus des imputations (triangle blanc), et un intervalle ([-]) qui contient le 95% des 1000 intensités d'implication obtenues. On constate les grandes proximités de ces triangles. Par des raisons de place dans cette communication, on n'a pris que les 4 variables ayant les plus grandes dispersions des intensités d'implication.

Le calcul de la matrice implicative est programmé avec une fonction définie dans l'annexe 2, tandis que le processus complet d'obtention des valeurs, intervalles et graphes est implémenté avec le programme de l'annexe 7.

8 Conclusions

L'imputation basée sur l'ASI se présente comme une alternative sérieuse à d'autres méthodes d'imputation, comme l'imputation par l'utilisation du mode et celle de la régression, en particulier quand les données manquantes sont « abondantes » dans le tableau. Les résultats obtenus ici et reproductibles à l'aide des petits programmes figurant dans les annexes semblent en effet créditer notre méthode d'analyse statistique implicative de qualités, non nécessairement attendues bien que visées. Ils montrent en toute hypothèse que la prise en compte, d'une part, des profils des sujets par leur proximité comportementale et, d'autre part, des relations implicatives induites, sous-jacentes entre les variables en est sans doute la source. Ce que ne font pas les autres méthodes connues. D'ailleurs, une simulation à partir d'un fichier aléatoirement simulé, ne présentant pas de structure interne a priori, nous a montré l'inanité de notre méthode de complétion. En revanche, la prise en compte d'un tableau réel structuré par les relations sujets-variables, éventuellement complété par l'introduction des variables supplémentaires descriptives, conduit à des résultats très encourageants vers une solution approchée du délicat problème posé par des données manquantes. D'autres situations ultérieurement examinées pourraient conforter l'hypothèse d'une qualité inductive de l'ASI exploitée à travers cette fois l'imputation de données se substituant à des absences de données.

On a donc vu que l'imputation multiple nous permettait de déduire des estimations des valeurs de l'intensité de l'implication entre les couples de variables (connues quand on a le tableau complet), et même des informations sur leur variabilité lorsque les données sont manquantes. Évidemment, le modèle utilisé pour effectuer l'imputation multiple a une influence sur les résultats obtenus, et c'est pour cette raison qu'il faut en justifier le choix **sémantiquement et épistémologiquement**.

CHIC 4.1 est un outil très adapté, pour sa facilité d'utilisation et la qualité des produits graphiques, pour décrire les résultats de l'ASI appliquée à n'importe quel ensemble de données. Toutefois, pour faciliter le développement théorique des aspects basés sur l'ASI (principalement ceux qui comportent la simulation intensive), le logiciel CHIC devrait intégrer un langage de commandes permettant de manipuler et d'exploiter ses propres résultats. En outre, il devrait étendre son système d'entrée/sortie à l'exportation des résultats, qui seraient traités par d'autres programmes statistiques, et à l'importation, pour permettre de poursuivre l'analyse implicative.

R (R, 2008) est un logiciel de licence libre (GPL) qui dispose d'un langage de programmation commun et intégrateur (peut-être pas le plus efficace à niveau informatique) dans la Communauté scientifique statistique internationale, et pour lequel les chercheurs contribuent à son expansion quotidienne à travers le monde. Les algorithmes programmés dans R sont des fonctions qui prennent un certain nombre d'arguments (des variables de type vecteur, matrice, tableaux de données, listes, etc.) et dont les résultats peuvent être gérés comme de nouvelles variables. Ainsi ces fonctions peuvent servir d'arguments pour de nouvelles fonctions, facilitant le flux d'analyses.

Par exemple, pour appliquer les algorithmes ici utilisés, il suffit d'ouvrir le logiciel R, copier-coller le code, facilité par les auteurs sous demande, écrire une ligne de code pour lire le tableau de données intéressant (par exemple dans la variable `mesdonnees`), et écrire une autre ligne

```
gras.na.input(x=mesdonnes, random=50000)
```

pour avoir le résultat des 50000 affectations aléatoires sur le tableau, puis de les traiter statistiquement à volonté. En plus, on pourrait changer les choix arbitraires qu'on a pris dans notre texte en reformulant les fonctions définies dans des buts particuliers. Si CHIC étend son système d'entrée / sortie, nous pensons qu'un nouveau design permettant l'interaction avec R serait très productif.

Remerciements

Les auteurs remercient les importants commentaires reçus de la part de Pilar Orús aussi que son soutien quotidien, qui a aidé à la matérialisation de la contribution présentée ici. Ils remercient également les reviewers anonymes du comité scientifique des rencontres ASI5, dont leurs commentaires ont permis l'amélioration de la version première de la contribution.

References

Couturier, R., Ag Almouloud, S. (2009). Historique et fonctionnalités de CHIC. In R. Gras, J. C. Regnier & F. Guillet (Eds.), *Analyse Statistique Implicative : une méthode d'analyse de données pour la recherche des causalités* (pp. 175–182). Paris: Cépadues.

Dondersa, A. R. T., van der Heijdenc, G. J. M. G., Stijnend, T., Moons, K. G. M. (2006). Review: A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, 59, 1087–1091

FUNDESTAP, Equipo de investigación (2010). *Estudio Sociológico de la población de Alcublas: Necesidades percibidas, carencias, capacidades y propuestas de mejora*. Valencia. Ayuntamiento de Alcublas

Gras, R. (2009). Problème de données manquantes dans un tableau numérique : une application de l'analyse statistique implicative. In R. Gras, J. C. Regnier & F. Guillet (Eds.), *Analyse Statistique Implicative : une méthode d'analyse de données pour la recherche des causalités* (pp. 175–182). Paris: Cépadues.

Gras, R., Ag Almouloud, S., Bailleul, M., Lahrer, A., Polo, M., Ratsimba-Rajohn, H., Totohasina, A. (1996). *Analyse Statistique Implicative : une méthode d'analyse de données pour la recherche des causalités*. Grenoble: La Pensée Sauvage.

Gras, R., Regnier, J. C., Guillet, F. (2009). *Analyse Statistique Implicative : une méthode d'analyse de données pour la recherche des causalités*. RNTI E-16. Paris: Cépaduès.

Gras, R., Suzuki, E., Guillet, F., Spagnolo, F. (2008). *Statistical implicative analysis: theory and applications*. Studies in Computational Intellingence 127. New York: Springer.

Jones, M. P. (1996). Indicator and stratification methods for missing explanatory variables in multiple linear regression. *Journal of the American Statistical Association*, 91(433), 222–230.

Little, L. J. A., Rubin, D. B. (1987). *Statistical analysis with missing data*. New York: J. Wiley & Sons.

Orús, P., Zamora, L., Gregori, P. (2009). *Teoría y aplicaciones de análisis estadístico implicativo: primera aproximación en lengua hispana*. ISBN: 978-84-692-3925-4. Castellón: Departamento de Matemáticas de la Universitat Jaume I.

R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, http://www.R-project.org.

Ratsimba-Rajohn, H. (2009). Guide d'utilisation des principales fonctionnalités du logiciel CHIC. In R. Gras, J. C. Regnier & F. Guillet (Eds.), *Analyse Statistique Implicative : une méthode d'analyse de données pour la recherche des causalités* (pp. 175–182). Paris: Cépadues.

Rubin, D. B. (1987). *Multiple imputation for non response in surveys*. New York: J. Wiley & Sons.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Monographs on Statistics and Applied Probability 72. Chapman & Hall.

Annexe 1

Fichier `auxiliary_na.R` : fonctions pour des opérations simples qui opèrent avec des données manquantes.

```
# sum discarding NA values
sum.na <- function( x ) {
  return( sum(x, na.rm=TRUE) )
}

# minimum discarding NA values
min.na <- function( x ) {
  return( min(x, na.rm=TRUE) )
}

# maximum discarding NA values
max.na <- function( x ) {
  return( max(x, na.rm=TRUE) )
}

# counting NA values in a vector
howmany.na <- function( x ) {
  return( sum(is.na(x))-sum(is.nan(x)) )
}
```

Annexe 2

Fichier `asi_intensity0.R` : fonction qui calcule la matrice implicative selon la théorie classique en utilisant le modèle de Poisson.

```
asi.intensity0 <- function(x) {
  # Arguments:
  #   x: binary data frame
  # Returns:
  #   res: implicative matrix under classical
  #        version and Poisson model
  n <- dim(x)[1]
  p <- dim(x)[2]
  x2 <- data.matrix(x)
  n.a.non.b <- t(x2) %*% (1-x2)
  n.a <- apply(X=x2, MARGIN=2, FUN='sum')
  n.non.b <- apply(X=1-x2, MARGIN=2, FUN='sum')
  l <- tcrossprod( n.a, n.non.b )/n
  res <- 1-ppois(q=n.a.non.b, lambda=l )
  diag(res) <- 0
  return( res )
}
```

Annexe 3

Fichier `other_na_input.R` : fonction qui réalise les deux types d'imputation selon le mode et en utilisant la régression.

```
other.na.input <- function(x, type='mode') {
  # Arguments:
  #   x: data frame with missing data (NA).
  #   type: the strategy for imputation. 'mode' uses the mode
of
  #          each variable. 'logreg' uses a logistic regres-
sion
  #         over the complete cases table.
  # Returns:
  #   y: data frame with imputed data.
  y <- x
  n <- dim(x)[1]
  p <- dim(x)[2]
  gaps <- which(is.na(x))
  if( length(gaps) > 0 ) {
    cg <- ((gaps-1) %/% n) + 1
    rg <- ((gaps-1) %% n) + 1
    ucg <- unique(cg)
    if( type=='mode' ) {
      for( var in ucg ) {
        a <- table(x[,var])
        mode        <-        as.numeric(attr(a,        'dim-
names')[[1]][which.max(a)])
        y[,var][is.na(x[ , var])] <- mode
      }
    }
    if( type=='logreg' ) {
      xcomp <- x[ complete.cases(x), ]
      if( dim(xcomp)[1] < 2 ) {
        y <- x
      } else {
        for( j in 1:length(gaps) ) {
          thiscol <- cg[j]
          thisrow <- rg[j]
          thisemptycols <- cg[ rg==thisrow ]
          subset <- rep(F, n)
          subset[ thisrow ] <- T
          newdata    <-    subset(x=y,    subset=subset,    se-
lect=(1:p)[-thisemptycols] )
          formula <- paste( names(x)[thiscol], ' ~ ',
                            paste(names(x)[  -thisemptycols],
collapse='+'),
                            collapse='')
          reglog  <-  glm(formula=formula,  data=xcomp,  fam-
ily='binomial')
          y[ rg[j], cg[j] ] <-
                floor(2/(1+exp(-(predict.glm(object=reglog,
newdata=newdata)))))
        }
      }
    }
  }
  return(y)
}
```

Annexe 4

Fichier `asi_option.R` : fonction complémentaire pour appliquer l'option de la deuxième approche de l'imputation basée sur l'ASI (qui prend en compte des implications). Ce code a une dépendance au code de l'annexe 1.

```
# auxiliar function for ASI correction
# to proposal for imputing missing data
source('auxiliary_na.R')

asi.option <- function(proposal=NULL, x=NULL, col.gap=NULL,
row.gap=NULL,
                       implic=NULL,    cohes=NULL,    thresh-
old=NULL) {
  # Arguments:
  #   proposal: a proposal for the imputation of the missing
data.
  #   x: the data frame for the imputation.
  #   col.gap: the column where the gap is.
  #   row.gap: the row where the gap is.
  #    implic: matrix of intensity implications of the known
part of 'x'.
  #   cohes: matrix of cohesions of the known part of 'x'.
  #     threshold: for considering or not the implicative
classes
  # Returns:
  #   res: the new proposal.
  res <- proposal
  diag(implic) <- 0.0
  class.antec <- implic[ ,col.gap] >= threshold
  class.consec <- implic[col.gap, ] >= threshold
  length.class.antec <- sum(class.antec)
  length.class.consec <- sum(class.consec)
  if( length.class.antec > 0 ) {
    subcohes <- cohes[ class.antec, class.antec ]
    subcohes[ lower.tri(x=subcohes, diag=TRUE) ] <- 1.0
    coh.class.antec         <-         prod(         subcohes
)^(1/choose(length.class.antec,2))
    # cohesion of class 'pères'
    implic.antec.max    <-    implic[    ,col.gap]==max(implic[
,col.gap])
  } else {
    coh.class.antec <- -Inf
  }
  if( length.class.consec > 0 ) {
    subcohes <- cohes[ class.consec, class.consec ]
    subcohes[ lower.tri(x=subcohes, diag=TRUE) ] <- 1.0
    coh.class.consec         <-         prod(         subcohes
)^(1/choose(length.class.consec,2))
    # cohesion of class 'fils'
    implic.consec.max           <-           implic[col.gap,
]==max(implic[col.gap, ])
    } else {
    coh.class.consec <- -Inf
  }
  if( coh.class.antec >= coh.class.consec ) {
    compare    <-    c(proposal,    as.numeric(x[row.gap,    im-
plic.antec.max]))
    res <- min.na(compare)
```

```
  }
  if( coh.class.antec < coh.class.consec ) {
    compare   <-   c(proposal,   as.numeric(x[row.gap,   im-
plic.consec.max]))
    res <- max.na(compare)
  }
  return(res)
}
```

Annexe 5

Fichier `gras_na_input.R` : fonction principale qui réalise l'imputation basée sur l'ASI. Ce code a une dépendance au code des annexes 1, 2 et 4.

```
gras.na.input <- function(x, asi=FALSE, implic=NULL, co-
hes=NULL,
                          threshold=NULL, random=FALSE) {
  # Arguments:
  #   x: data frame with missing data (NA).
  #   asi: TRUE if the implications are to be taken into ac-
count.
  #    implic: implicative matrix of the complete cases of
'x'.
  #           Necessary when ASI=TRUE.
  #   cohes: matrix of cohesions of the known part of 'x'.
  #          Necessary when ASI=TRUE, it is computed from the
  #          provided implicative matrix.
  #    threshold: value for considering only the 'important'
implicative
  #              classes. Necessary when ASI=TRUE.
  #   random: if FALSE, the imputation is deterministic,
  #           otherwise it determines the number of random
  #           imputations.
  # Returns:
  #   y: data frame with imputed data. If random=FALSE,
  #      'y' has the same structure as 'x'. Otherwise, 'y'
  #       is an array of dimensions c(dim(x)[1], dim(x)[2],
random), and
  #      'y[,,i]' is the i-th random imputation.
  source('auxiliary_na.R')
  source('asi_option.R')
  source('asi_intensity0.R')
  if( asi==TRUE ) {
    if( is.null(cohes) ) {
      E <- - implic * log2(implic) - (1 - implic) * log2( 1-
implic )
      E[ is.nan(E) ] <- 0
      cohes <- sqrt(1-E^2)
      cohes[ implic < 0.5 ] <- 0
      diag(cohes) <- 1
    }
  }
  warning <- character(0) # vector with warning messages,
initially empty
  n <- dim(x)[1] # number of rows of data matrix 'x'
  p <- dim(x)[2] # number of columns of data matrix 'x'
  gaps <- which( is.na(x) ) # indices of NAs of 'x' (read
columnwise)
  row.na <- which( !complete.cases(x) ) # rows of 'x' with
some NA
  x.mask <- is.na(x) # mask for 'available' (nonmissing) data
in 'x'
  if( random==FALSE ) {
    y <- x # y will be the copy of 'x' with the imputations
  } else {
    # or an array of 'random' copies of 'x',
    # result of the multiple imputation
    if( is.null(dimnames(x)) ) {
```

```
      y <- array(data=data.matrix(x), dim=c(n,p,random),
                 dimnames=list(NULL, NULL, 1:random))
    } else {
      y <- array(data=data.matrix(x), dim=c(n,p,random),
                 dimnames=c(dimnames(x), list(1:random)))
    }
  }
  if( length(row.na)==0 ) {
    cat('Your datasheet has no missing data\n')
    return(x)
  }
  for( row.gap in row.na ) {    # For any row in 'x' showing
gap(s)...
    # id.x: individual in matrix form
    id.x <- x[ rep(x=row.gap, times=n), ]
    id.x[ x.mask ] <- NA
    mask <- is.na(id.x)
    # id.y: rest of individuals
    id.y <- x
    id.y[ mask ] <- NA
    # id.x.dot: margin for id.x
    id.x.margin <- apply( X=id.x, MAR=1, FUN='sum.na' )
    id.x.howmany.non.na <-
           array(data=    p    -    apply(    X=id.x,    MAR=1,
FUN='howmany.na' ), dim=c(n,p))
    id.x.dot   <-   matrix(data=rep(x=id.x.margin,   times=p),
ncol=p)
    id.x.dot[ mask ] <- NA
    # id.y.dot: margin for the rest of individuals (id.y)
    id.y.margin <- apply( X=id.y, MAR=1, FUN='sum.na' )
    id.y.howmany.non.na <-
           array(data=    p    -    apply(    X=id.y,    MAR=1,
FUN='howmany.na' ), dim=c(n,p))
    id.y.dot   <-   matrix(data=rep(x=id.y.margin,   times=p),
ncol=p)
    id.y.dot[ mask ] <- NA
    # n.dot: margin for variables
    n.margin <- apply( X=id.y, MAR=2, FUN='sum.na' )
    n.dot  <-  matrix(data=rep(x=n.margin,  times=n),  ncol=p,
byrow=TRUE)
    n.dot[ mask ] <- NA
    # N.dot: total sum of variable margins
    N.margin <- apply( X=n.dot, MAR=1, FUN='sum.na' )
    N.dot <- matrix(data=rep(x=N.margin, times=p), ncol=p)
    N.dot[ mask ] <- NA
    id.x.profile <- id.x/id.x.dot
    id.x.profile.mask.nan                                    <-
is.nan(data.matrix(id.x.profile))
    id.x.profile[ id.x.profile.mask.nan ] <-
                                (1/id.x.howmany.non.na)[
id.x.profile.mask.nan ]
    # solves the indeterminations 0/0
    id.y.profile <- id.y/id.y.dot
    id.y.profile.mask.nan                                    <-
is.nan(data.matrix(id.y.profile))
    id.y.profile[ id.y.profile.mask.nan ] <-
                                (1/id.y.howmany.non.na)[
id.y.profile.mask.nan ]
    # solves the indeterminations 0/0
```

```
    # std.diff: standarized difference between id.x and id.y
    std.diff <- (id.x.profile - id.y.profile)^2/(n.dot/N.dot)
    std.diff[ is.nan(std.diff) ] <- 0.0     # solves the in-
determinations 0/0
    # dist: vector of all distances between id.x and id.y
    dist <- apply( X=std.diff, MAR=1, FUN='sum.na' )
    # k: number of common variables between id.x and id.y
    k <- apply(X=!mask, MAR=1, FUN='sum')
    # delta: vector of corrected distances between id.x and
id.y
    delta <- (p-k)/p * dist
    col.gap <- which( is.na( x[row.gap, ] ) )
    # the index of columns having missing data in the row
    if( random==FALSE ) {
      closer.id   <-   order(x=delta,   y=-k,   z=-id.y.margin,
w=1:n)
      # closer.id: Vector with individuals ordered
      #                   increasingly by distance 'delta' to
the individual
      #                 of the current missing data.
      #                   Ties are broken as detailed in Gras
(2009)
      for( gap in col.gap ) {
        closer.id.gap <- closer.id[ !is.na(x[closer.id, gap])
]
        # leave out individuals with NA since they will not
help
        if( length(closer.id.gap)==0 ) {
          warning <- c(warning,
                       paste('Imputation at missing data [',
row.gap, ', ',
                             col.gap,
                             '] could not be completed by a
lack of data.\n',
                             sep=''))
          on.exit(cat(warning))
          proposal <- NA
        } else {
          proposal <- x[ closer.id.gap[1], gap ]
        }
        if( asi==TRUE ) {
          # ASI correction to the proposed imputation
          proposal   <-   asi.option(proposal=proposal,   x=x,
col.gap=gap,
                                     row.gap=row.gap,         im-
plic=implic,
                                     cohes=cohes,         thresh-
old=threshold)
        }
        y[ row.gap, gap ] <- proposal
      }
    } else { # in the case of multiple imputation
      nsamples <- random
      for( gap in col.gap ) {
        sampleable <- !is.na(x[, gap])
        if( length( which( sampleable ) )==0 ) {
          proposal <- NA
          warning <- c(warning,
```

```
                    paste('Imputation at missing data [',
row.gap, ', ',
                          col.gap,
                          '] could not be completed by a
lack of data.\n',
                          sep=''))
          on.exit(cat(warning))
        } else {
          ids.delta0 <- which( delta==0 & sampleable )
          alpha <- 2 # parameter at choice
          prob <- exp( -alpha*delta )
          # weights as a function of the distance 'delta'
          prob[ !sampleable ] <- 0.0
          # special '0' weight for non interesting individu-
als
          prob[ ids.delta0 ] <- 2
          # special 'extra' weigth for 0-distance individuals
          id.proposal <- sample( x=1:n, size=nsamples, re-
place=TRUE, prob=prob )
          proposal <- x[ id.proposal, gap ]
        }
        if( asi==TRUE ) {
          # ASI correction to the proposed imputation
          proposal <- asi.option(proposal=proposal, x=x,
col.gap=gap,
                                 row.gap=row.gap,         im-
plic=implic,
                                 cohes=cohes,         thresh-
old=threshold)
        }
        y[ row.gap, gap, ] <- proposal
      }
    }
  }
  return(y)
}
```

Annexe 6

Code qui permet la comparaison des méthodes d'imputation exposées dans la section 5. Ce code a une dépendance au code des annexes 2, 3 et 5.

```
x <- read.csv2(file='alcublas.csv')
# the dataset. Please use your available dataset instead
n <- dim(x)[1]
p <- dim(x)[2]
xm <- data.matrix(x)
source('asi_intensity0.R')
source('gras_na_input.R')
source('other_na_input.R')
# we force the "missingness" of n0 entries of the dataset
n0 <- c(floor(0.05*n*p), #  5% of missing data
        floor(0.1*n*p),  # 10% of missing data
        floor(0.2*n*p) ) # 20% of missing data
# then we will proceed to imputation of missing data under
# three methods: ASI, MODE, REG
#   ASI: following Gras (2009)
#   MODE: imputation of the mode at each variable
#   REG: take all the complete columns as regressors
#        and predict missing data only with complete cases
# and we will repeat the process in order to approximate
# the expected rate of succes of each method
N <- 1000 # number of iterations
taux <- array(data=0, dim=c(N, length(n0), 3),
              dimnames=list(NULL,   NULL,   c('ASI',   'REG',
'MODE')) )
# for storing the rates of correct imputations
set.seed(20100720)
for( j in 1:length(n0) ) { # for each scénario 5%, 10% and
20%
  for( i in 1:N ) { # for each repetition
    gap <- sample(x=n*p, size=n0[j])    # choose gaps at ran-
dom
    x0v <- as.numeric( data.matrix(x) )
    x0v[ gap ] <- NA
    x0m   <-   matrix(data=x0v,   ncol=p,   dimnames=list(1:n,
names(x)) )
    # creates the uncomplete dataset
    x0 <- data.frame(x0m)
  # 3. Complete the table under (1) ASI, (2) MODE, (3) REG
    asi <- FALSE
    if( asi==TRUE ) {
      x0cmp <- x0[ complete.cases(x0), ]
      impl <- asi.intensity0(x0cmp)
      E <- - impl * log2(impl) - (1 - impl) * log2( 1-impl )
      E[ is.nan(E) ] <- 0
      coh <- sqrt(1-E^2)
      coh[ impl < 0.5 ] <- 0
      diag(coh) <- 1
      x1  <-  gras.na.input(x0,  asi=TRUE,  implic=impl,  co-
hes=coh, threshold=0.75)
    } else {
      x1 <- gras.na.input(x0)
    }
    x2 <- other.na.input(x0, type='mode')
    x3 <- other.na.input(x0, type='logreg')
```

```
  # 4. Compute the rate of correct imputations of all the
gaps
    taux[i,j,'ASI']  <-  sum(  xm[gap]==data.matrix(x1)[gap]
)/n0[j]
    taux[i,j,'MODE']  <-  sum(  xm[gap]==data.matrix(x2)[gap]
)/n0[j]
    taux[i,j,'REG']  <-  sum(  xm[gap]==data.matrix(x3)[gap]
)/n0[j]
  }
}
# 7. Summarize the results on the observed rates
par(mfcol=c(1,3))
for( j in 1:length(n0) ) {
  gaps.ok <- 100*taux[,j,c('ASI', 'REG', 'MODE')]
  boxplot(as.data.frame(gaps.ok),  ylab="Taux  d'effectivité
(%)")
}
```

Annexe 7

Code qui permet d'analyser les estimations des intensités d'implication par l'imputation multiple basée sur l'ASI décrite dans la section 7. Ce code a une dépendance au code de l'annexe 6.

```
x <- read.csv2(file='alcublas.csv')
# the dataset. Please use your available dataset instead
n <- dim(x)[1]
p <- dim(x)[2]
set.seed(20100720)
gap <- sample(x=n*p, size=n0)
x0v <- as.numeric( data.matrix(x) )
x0v[ gap ] <- NA
x0m <- matrix(data=x0v, ncol=p, dimnames=list(1:n, names(x))
)
# creates the uncomplete dataset
x0 <- data.frame(x0m)
asi <- FALSE
nsamples <- 10000
y <- gras.na.input(x=x0, random=nsamples)
all.implic.alcublas.asi <- array(data=0, dim=c(p, p, nsam-
ples),
                    dimnames=list(names(x),         names(x),
1:nsamples))
for( i in 1:nsamples ) {
  all.implic.alcublas.asi[,, i] <-
    asi.intensity0( data.matrix(y[,,i]) )
  # computes implicative matrix (classical, Poisson)
  # it is unfeasible to do it all with CHIC 4.1
}
# all.implic.alcublas.asi[i,j,k] is the intensity of implica-
tion
# i => j for the k-th inputation.
q2_5 <- function(x) { return( quantile(x, prob=0.025) ) }
q97_5 <- function(x) { return( quantile(x, prob=0.975) ) }
p2_5.alcublas.asi <- apply( X=all.implic.alcublas.asi,
                                MAR=c(1,2), FUN='q2_5' )
p97_5.alcublas.asi <- apply( X=all.implic.alcublas.asi,
                                MAR=c(1,2), FUN='q97_5' )
mean.alcublas.asi <- apply( X=all.implic.alcublas.asi,
                                MAR=c(1,2), FUN='mean' )
# We show only the most variable implications
choose      <-      c('AT_FUNCMUN_53',      'RAPGESTMUN_54',
'ATP_ADOLES_84', 'ATP_JUVENT_85')
# please put here the names of the chosen variables of your
dataset instead
implic.ref <- asi.intensity0(x)[choose, choose]
# this is the true (= complete data) implicative submatrix
# Graphical representation
par( mfrow=c(length(choose),length(choose)), mar=c(2,2,1,1))
for( antec in choose ) {
  for( consec in choose ) {
    if(antec==consec) {
      plot(x=0, y=0, col='white', xaxt='n', yaxt='n')
      text(x=0, y=0, antec, cex=1.5)
    } else {
      samp <- all.implic.alcublas.asi[antec, consec, ]
      tab <- table(samp)
```

```
      if( length(tab) < 10 ) {
        tab.labs <- names(table(samp))
        tab2 <- strtrim(tab.labs,5)
        tabfreq <- tab/length(samp)
        plot(x=as.numeric(tab.labs),
             y=tabfreq, ylim=c(0, max(tabfreq)), xlim=c(0,1),
type='h')
        points(x=as.numeric(tab.labs), y=tabfreq, pch=1)
        points(x=implic.ref[antec, consec],
               y=0, pch=17, cex=2)
        points(x=implic.ref[antec, consec],
               y=0, pch=17, cex=2)
        points(x=mean.alcublas.asi[antec, consec],
               y=0, pch=2, cex=1.7)
        points(x=p2_5.alcublas.asi[antec, consec],
               y=0, pch='[', cex=2, col='red')
        points(x=p97_5.alcublas.asi[antec, consec],
               y=0, pch=']', cex=2, col='red')
        points(x=c(p2_5.alcublas.asi[antec, consec],
                   p97_5.alcublas.asi[antec, consec]),
               y=c(0,0), type='l', lwd=2, col='red')
      }
      else {
        hist(x=samp, main='', sub='', xlim=c(0,1))
        points(x=implic.ref[antec, consec],
               y=0, pch=17, cex=2)
        points(x=mean.alcublas.asi[antec, consec],
               y=0, pch=2, cex=1.7)
        points(x=p2_5.alcublas.asi[antec, consec],
               y=0, pch='[', cex=2, col='red')
        points(x=p97_5.alcublas.asi[antec, consec],
               y=0, pch=']', cex=2, col='red')
        points(x=c(p2_5.alcublas.asi[antec, consec],
                   p97_5.alcublas.asi[antec, consec]),
               y=c(0,0), type='l', lwd=2, col='red')
        box()
      }
    }
  }
}
```