

DESCRIPTION DES FONCTIONS
DU SYSTÈME BANFF POUR LA
VÉRIFICATION ET L'IMPUTATION

Banff

Version 2.07

Équipe de soutien de Banff
Mars 2017



Statistique
Canada Statistics
Canada

Canada

Afin de faire connaître aux utilisateurs de Banff les diverses possibilités qu'il offre, une variété de documents, incluant celui-ci, a été préparée par Statistique Canada. Le lecteur peut consulter la section des références pour trouver les coordonnées de ceux-ci.

Le présent document décrit la méthodologie utilisée dans les fonctions de Banff. Il devrait intéresser toute personne qui désire étudier la logique sous-jacente de chaque fonction de Banff de manière à mieux évaluer les différentes options qui lui sont offertes.

Des copies de ce document peuvent être obtenues auprès de :

Statistique Canada
Division des méthodes d'enquêtes auprès des entreprises
Section de l'Assurance de la qualité et Systèmes généralisés
17^{ième} étage, Immeuble R.-H.- Coats
Ottawa, Ontario
K1A 0T6

This document is also available in English.

TABLE DES MATIÈRES

1. INTRODUCTION	1 - 1
Contexte	1 - 1
Indépendance des procédures SAS	1 - 1
L'énonce BY dans Banff	1 - 2
Objet du présent document	1 - 2
Autres documents disponibles	1 - 2
Structure du document	1 - 3
Proc Verifyedits - Spécification et analyse des règles de vérification.....	1 - 4
Proc Editstats - Statistiques sommaires sur les règles de vérification	1 - 4
Proc Outlier - Détection des valeurs aberrantes.....	1 - 4
Proc Errorloc - Localisation des erreurs	1 - 4
Proc Deterministic - Imputation déterministe.....	1 - 5
Proc Donorimputation - Imputation par enregistrement donneur.....	1 - 5
Proc Estimator - Imputation par estimateur	1 - 5
Proc Prorate - Pro-Rating.....	1 - 5
Proc Massimputation - Imputation massive.....	1 - 5
Mises à jour du présent document	1 - 6
2. PROC VERIFYEDITS - SPÉCIFICATION ET ANALYSE DES RÈGLES DE VÉRIFICATION	2 - 1
2.1 SPÉCIFICATION DES RÈGLES DE VÉRIFICATION	2 - 1
Objet	2 - 1
Les règles de vérification spécifiées dans Banff.....	2 - 1
La forme canonique	2 - 2
Les règles de vérification non linéaires	2 - 2
Règles de vérification pour le traitement des valeurs négatives	2 - 3
Les groupes de règles de vérification et les groupes de données.....	2 - 3
2.2 VÉRIFICATION DES RÈGLES	2 - 6
Objet	2 - 6
Définitions	2 - 6
Groupes de règles de vérification cohérents et incohérents	2 - 7
Règles redondantes; règles strictes mais non restrictives	2 - 8
Égalités cachées	2 - 9
Limites supérieures et inférieures; variables déterministes	2 - 10
2.3 GÉNÉRATION DES POINTS EXTRÉMAUX	2 - 11
Objet	2 - 11
Description de la méthode employée.....	2 - 11
Exemples de points extrémaux	2 - 12
2.4 GÉNÉRATION DES RÈGLES DE VÉRIFICATION IMPLICITES	2 - 13
Objet	2 - 13
Description de la méthode employée.....	2 - 13
Exemples de règles de vérification implicites	2 - 13

3. PROC EDITSTATS - STATISTIQUES SOMMAIRES SUR LES RÈGLES DE VÉRIFICATION	3 - 1
Objet	3 - 1
Description de la méthode employée	3 - 1
Exemple de création de codes d'état pour chaque règle	3 - 2
Exemples de tableaux	3 - 3
TABLEAU 1-1	3 - 3
TABLEAU 1-2	3 - 4
TABLEAU 1-3	3 - 5
TABLEAU 2-1	3 - 5
TABLEAU 2-2	3 - 6
Utilisation des statistiques sommaires sur les règles de vérification	3 - 7
Règles de vérification pour le traitement des valeurs négatives	3 - 8
4. PROC OUTLIER - DÉTECTION DES VALEURS ABERRANTES	4 - 1
Objet	4 - 1
Options offertes dans Proc Outlier.....	4 - 1
Description de la méthode Hidirogloou et Berthelot utilisant des données courantes	4 - 2
Exemple d'application de la méthode Hidirogloou et Berthelot utilisant des données courantes	4 - 3
Description de la méthode Hidirogloou et Berthelot utilisant des ratios	4 - 5
Description de la méthode Hidirogloou et Berthelot utilisant des tendances historiques	4 - 6
Exemple d'application de la méthode Hidirogloou et Berthelot utilisant des tendances historiques	4 - 6
Comparaison entre la méthode Hidirogloou et Berthelot utilisant des données courantes et utilisant des tendances historiques	4 - 10
Description de la méthode de l'écart-sigma utilisant des données courantes.....	4 - 11
Premier exemple d'application de la méthode de l'écart-sigma utilisant des données courantes	4 - 14
Deuxième exemple d'application de la méthode de l'écart-sigma utilisant des données courantes	4 - 15
Description de la méthode de l'écart-sigma utilisant des ratios.....	4 - 16
Description de la méthode de l'écart-sigma utilisant des tendances historiques	4 - 17
Exemple d'application de la méthode de l'écart-sigma utilisant des tendances historiques ...	4 - 17
5. PROC ERRORLOC - LOCALISATION DES ERREURS	5 - 1
Objet	5 - 1
Description de la méthode employée	5 - 1
Les solutions multiples	5 - 3
Exemple de localisation des erreurs.....	5 - 3
L'algorithme de Chernikova	5 - 5
Règles de vérification pour le traitement des valeurs négatives	5 - 6
L'utilisation de poids dans la procédure de localisation des erreurs	5 - 6
Limite du nombre de champs visés par la solution.....	5 - 7
Aucune solution trouvée - Imputation manuelle requise	5 - 7
Aucune solution trouvée - Délai dépassé.....	5 - 7
Autres sources de champs à imputer.....	5 - 7

6. PROC DETERMINISTIC - IMPUTATION DÉTERMINISTE	6 - 1
Objet	7 - 1
Description de la méthode employée.....	6 - 1
Exemple d'imputation déterministe	6 - 1
Règles de vérification pour le traitement des valeurs négatives	6 - 3
7. PROC DONORIMPUTATION - IMPUTATION PAR ENREGISTREMENT DONNEUR.....	7 - 1
Objet	7 - 1
Définitions	7 - 1
Exemple de classement des enregistrements	7 - 2
Règles de vérification pour le traitement des valeurs négatives	7 - 4
Imputation massive	7 - 4
7.1 PRÉPARATION DE L'IMPUTATION PAR ENREGISTREMENT DONNEUR	7 - 5
Objet	7 - 5
Données précédemment imputées dans des enregistrements donneurs.....	7 - 5
Autres exclusions de la population d'enregistrements donneurs	7 - 5
Critères pour l'imputation par enregistrement donneur	7 - 6
7.2 DÉTERMINATION DES CHAMPS D'APPARIEMENT	7 - 7
Objet	7 - 7
Champs d'appariement spécifiés par l'utilisateur ou champs d'appariement obligatoires.....	7 - 7
Description de la méthode employée.....	7 - 8
Exemple de détermination des champs d'appariement	7 - 8
Exemple d'un cas où il n'y a pas de champ d'appariement	7 - 10
7.3 TRANSFORMATION DES CHAMPS D'APPARIEMENT	7 - 11
Objet	7 - 11
Description de la méthode employée.....	7 - 11
Exemple de transformation des valeurs	7 - 11
Calcul de la distance	7 - 12
Exemple du calcul de la distance	7 - 12
7.4 EXÉCUTION DE L'IMPUTATION PAR ENREGISTREMENT DONNEUR	7 - 13
Objet	7 - 13
Description de la méthode employée.....	7 - 13
Construction de l'arbre k-d.....	7 - 13
Exemple de construction d'un arbre	7 - 14
Enregistrements receveurs comportant des champs d'appariement - Parcours de l'arbre	7 - 17
Enregistrements receveurs ne comportant aucun champ d'appariement	7 - 19
Règles de vérification post-imputation	7 - 20
8. PROC ESTIMATOR - IMPUTATION PAR ESTIMATEUR.....	8 - 1
Objet	8 - 1
Types d'algorithmes	8 - 1
Description de la méthode employée.....	8 - 1

Description des algorithmes pré-définis dans Banff.....	8 - 2
Algorithmes définis par l'utilisateur.....	8 - 6
Moyennes pondérées et non pondérées	8 - 7
Variable de variance du modèle dans les régressions linéaires	8 - 7
Terme d'erreur aléatoire dans les régression linéaires.....	8 - 7
Calcul des paramètres d'une régression linéaire.....	8 - 8
Exclusion dans le calcul des paramètres.....	8 - 10
Critères relatifs au calcul des paramètres	8 - 11
Calcul des estimateurs utilisés pour l'imputation - Première exécution de la procédure Proc Estimator.....	8 - 11
Calcul des estimateurs utilisés pour l'imputation - Deuxième exécution de la procédure Proc Estimator, premier estimateur	8 - 13
Calcul des estimateurs utilisés pour l'imputation - Deuxième exécution de la procédure Proc Estimator, deuxième estimateur	8 - 14
 9. PROC PRORATE - AJUSTEMENT AU PRORATA	9 - 1
Objet	9 - 1
Description de la méthode employée	9 - 1
Contrôle de la syntaxe des règles de vérification	9 - 2
Algorithme d'ajustement au prorata	9 - 2
Algorithme de la méthode de base.....	9 - 2
Algorithme de la méthode « scaling »	9 - 3
Algorithme d'arrondissement	9 - 3
Exemple	9 - 4
Limites de variation	9 - 5
 10. PROC MASSIMPUTATION - IMPUTATION MASSIVE	10 - 1
Objet	10 - 1
Champs d'appariement.....	10 - 1
Autres paramètres	10 - 2
Description de la méthode employée.....	10 - 2
 11. RÉFÉRENCES	11 - 1
Annexe A - Calcul des médianes et des quartiles.....	A - 1
Annexe B - Algorithmes pré-définis dans Banff.....	B - 1

1. INTRODUCTION

Contexte

Le système de vérification et d'imputation Banff est un ensemble de procédures SAS spécialisées, mis au point par Statistique Canada, qui peuvent être utilisées chacune indépendamment, ou ensemble pour répondre aux besoins de vérification et d'imputation d'une enquête donnée. On suppose que les données qui sont traitées par le système Banff sont numériques et continues. De plus, les règles de vérification utilisées dans Banff doivent être exprimées sous forme linéaire. L'utilisateur peut choisir d'accepter ou rejeter les données négatives pour chaque exécution d'une procédure. Les exceptions sont la méthode des tendances historiques et la méthode des ratios dans la procédure de détection des valeurs aberrantes, qui n'acceptent que les données positives. On suppose également qu'une certaine vérification préliminaire a été faite à l'étape de saisie des données, et que le suivi des répondants est terminé.

Le système Banff remplace le Système généralisé de vérification et d'imputation (SGVI). La méthodologie actuellement utilisée dans Banff est presque identique à celle du SGVI. Toutefois, les deux systèmes présentent plusieurs différences importantes. En premier lieu, Banff est basé sur l'architecture SAS, alors que le SGVI employait Oracle comme structure sous-jacente des bases de données. En deuxième lieu, chaque procédure SAS dans Banff est indépendante des autres, tandis que les modules du SGVI étaient inter-reliés. En troisième lieu, non seulement Banff est disponible dans l'environnement UNIX comme l'était le SGVI, il est également disponible dans l'environnement Windows sur un ordinateur personnel. En raison de ces différences, l'utilisation de Banff est intrinsèquement beaucoup plus simple et plus souple que l'était celle du SGVI.

Indépendance des procédures SAS

L'indépendance des procédures SAS dans Banff donne à l'utilisateur beaucoup de liberté et de souplesse. Toutefois, cette indépendance accroît la responsabilité de l'utilisateur, qui doit s'assurer que les données d'entrée sont de bonne qualité et que les données de sortie sont interprétées et appliquées correctement. Dans le SGVI, les tables de données connexes sont mises à jour automatiquement par le système après chaque étape du traitement, car tous les modules sont reliés entre elles. En d'autres mots, les données initiales sont tout simplement écrasées par les nouvelles données. Cela rend la tâche difficile pour l'utilisateur, car si on veut retracer l'historique des changements apportés aux données, il faut créer manuellement des sauvegardes de ces tables. Il y a toutefois un aspect positif : comme les modules sont reliés entre eux alors le transfert des données d'un module à un autre est supposé correct, si le système fonctionne sans problème.

Dans Banff, chaque procédure accepte des données d'entrée indépendantes, fournies par l'utilisateur ou par une autre procédure Banff. Si les données d'entrée fournies par l'utilisateur proviennent de l'extérieur du système, il incombe à l'utilisateur de s'assurer de la qualité des données d'entrée. Banff tentera de traiter tout ce qu'on lui fournit.

De plus, chaque procédure produit ses propres données de sortie. Pour ce qui est des codes d'état des champs, le format des ensembles de données SAS est très similaire d'une procédure à l'autre, mais les codes seront différents. Les données de sortie produites par les procédures Banff contiennent uniquement les données qui ont été changées, par rapport aux données d'entrée. Par conséquent, il incombe à l'utilisateur d'incorporer ces changements dans ses données initiales à moins de travailler avec le processeur Banff (voir le *Guide de l'usager du processeur Banff 2.07*).

L'énoncé BY dans Banff

Tout comme les procédures SAS normales, les procédures Banff peuvent traiter les données par groupes BY. Par exemple, plutôt que de traiter des ensembles de données distincts pour chacun des groupes industriels, un utilisateur peut inclure tous les groupes industriels dans un même ensemble de données, et Banff traitera chacun de ces groupes indépendamment selon la variable BY qui identifie le groupe industriel.

Objet du présent document

Ce document vise à décrire les méthodes utilisées dans chacune des procédures Banff, à justifier d'un point de vue méthodologique le choix de ces procédures et à illustrer l'application de chaque procédure avec des exemples simples. Le document présente également les avantages et les inconvénients des paramètres associés à chaque procédure et donne aussi des exemples de nombreux cas particuliers en exposant la manière de les traiter dans Banff. Les méthodologues et les spécialistes du sujet pourraient utiliser ce guide pour comprendre la logique sous-jacente de Banff, ce qui leur permettra de mieux évaluer les différentes options de vérification et d'imputation.

Il n'est pas nécessaire de lire dans l'ordre les sections du document. La lectrice ou le lecteur intéressé par une procédure particulière peut consulter dès maintenant cette section, sans lire les sections précédentes.

Autres documents disponibles

Divers guides d'utilisation ont été écrits pour chacune des procédures SAS dans Banff. Ces guides décrivent la syntaxe, les paramètres pouvant être spécifiés et les options existantes. Des exemples sont aussi présentés. Ces guides sont tous inclus dans le document *Guide de l'usager des procédures de Banff 2.07* (Équipe de soutien de Banff, 2017) disponible auprès de l'équipe de soutien de Banff.

Banff peut être utilisé dans SAS ou, dans SAS Enterprise Guide grâce aux tâches Banff ou, en utilisant le processeur Banff. Le *Guide de l'usager du processeur Banff 2.07* (Équipe de soutien de Banff, 2017) explique comment utiliser le processeur Banff.

Le document *Banff Tutorial 2.06* (Équipe de soutien de Banff, 2014), qui s'applique également aux deux versions de Banff 2.06 et 2.07, fournit à l'utilisateur des données fondées sur une enquête réelle menée à Statistique Canada. Le manuel d'accompagnement présente étape par étape toutes les fonctions du système. Ce guide d'initiation devrait intéresser toute personne qui exploitera effectivement Banff ou qui préfère se familiariser avec le système dans un cadre « pratique ».

Morabito et Shields (1992) présentent des conseils, dans le *GEIS Applications User's Guide*, sur la façon d'améliorer les règles de vérification, de définir les paramètres appropriés et de personnaliser les procédures, en fonction des besoins d'une application particulière. Ce document se veut un guide de référence, qui sera utilisé pendant la spécification détaillée de la phase de vérification et d'imputation d'une enquête. Bien que ce guide ait été écrit pour le SGVI, un grand nombre des idées et des conseils qui y figurent s'appliquent également dans la plupart des cas au système Banff.

Structure du document

La figure 1.1 illustre les fonctions de base dans Banff, et les sections correspondantes du document qui traitent de chaque procédure SAS. En outre, la section 11 contient des références et les annexes présentant en détail la méthode utilisée dans Banff pour le calcul des médianes et des quartiles et pour l'imputation par estimateurs. Chaque section est brièvement décrite après la figure 1.1.

Les procédures Banff sont présentées dans l'ordre qu'un utilisateur pourrait les appliquer, s'il effectuait toutes les étapes de vérification et d'imputation dans le cadre d'une enquête type. **Toutefois, on doit garder à l'esprit que chaque procédure est une entité indépendante.** Ainsi, l'utilisateur peut sélectionner uniquement les procédures qu'il désire appliquer au traitement des données d'une enquête. Le diagramme suivant est néanmoins instructif, car il illustre l'enchaînement des procédures, depuis les étapes préliminaires de vérification jusqu'à la fin de l'imputation.

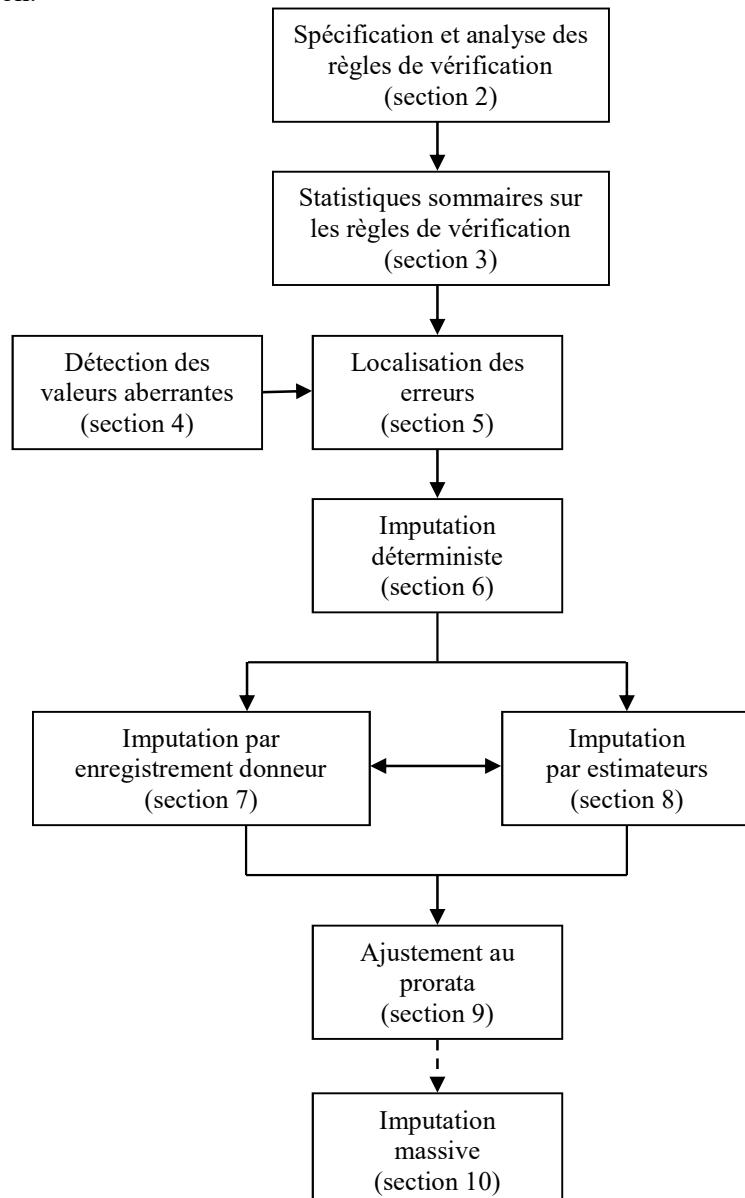


Figure 1.1 Ordre logique des procédures Banff, dans le contexte des traitements des données d'une enquête. Les numéros de section renvoient aux sections du présent document.

PROC VERIFYEDITS - Spécification et analyse des règles de vérification (section 2)

Quand on utilise les fonctions de vérification dans Banff, la première étape utile consiste à analyser les relations auxquelles doit satisfaire un enregistrement pour être admissible. Ces relations, appelées règles de vérification, revêtent une grande importance pour la bonne exécution de chaque procédure et pour la qualité des données produites par Banff. On peut établir ces conditions à partir de l'analyse du questionnaire, de l'analyse des données existantes et d'une connaissance du domaine spécialisé. Nous décrivons à la sous-section 2.1 (« **Spécification des règles de vérification** ») les règles de vérification valides dans Banff, ainsi que la formation des groupes de règles de vérification et des groupes de données. Nous décrivons aussi à la section 2 plusieurs procédures (**2.2 – Vérification des règles**, **2.3 – Génération des points extrêmaux** et **2.4 – Génération des règles de vérification implicites**) qui aident l'utilisateur à comprendre les conditions spécifiées et à s'assurer que le groupe de règles de vérification représente de façon exacte les conditions auxquelles les données doivent satisfaire. À cette étape, ce sont les règles de vérification elles-mêmes qui sont analysées. L'utilisateur ne précise pas la mesure à prendre lorsqu'un enregistrement ne satisfait pas aux conditions spécifiées; de fait, les données fournies par les répondants ne sont pas utilisées dans ces procédures.

PROC EDITSTATS – Statistiques sommaires sur les règles de vérification (section 3)

Si on dispose de données historiques ou préliminaires, on peut obtenir un aperçu des taux de rejet en produisant les **statistiques sommaires sur les règles de vérification** décrites à la section 3. Cette procédure est la première dans laquelle les règles de vérification linéaires sont appliquées aux données fournies par les répondants. Les cinq tableaux de statistiques sommaires sur les règles de vérification indiquent le nombre de fois où chaque code d'état et chaque code d'état global (PASS, MISS ou FAIL) a été attribué pour chaque variable et pour chaque règle ainsi que la répartition des enregistrements selon chacun des codes d'état pour un nombre donné de règles. Cette procédure ne fournit aucun renseignement concernant les champs devant être modifiés.

Les renseignements contenus dans les tableaux de statistiques sommaires sur les règles de vérification peuvent servir à améliorer les règles ou à estimer le nombre d'enregistrements qui seront rejettés durant l'exécution de la procédure de localisation des erreurs (et, par conséquent, à estimer le coût d'exécution sur l'ordinateur). Cette procédure peut aussi être exécutée après qu'une partie ou la totalité de l'imputation aura été faite afin d'évaluer le résultat du processus d'imputation.

PROC OUTLIER – Détection des valeurs aberrantes (section 4)

Cette procédure propose deux méthodes pour repérer les valeurs aberrantes, une décrite par Hidiroglou et Berthelot (1986) et la méthode de l'écart-sigma développée à Statistique Canada dans les années 1990. Les règles appliquées par cette procédure consistent à comparer les valeurs prises par certaines variables d'un enregistrement, plutôt que de comparer certains champs à l'intérieur de chaque enregistrement, comme dans le cas des règles de vérification linéaires. La détection des valeurs aberrantes peut aussi être fondée sur la variation de la valeur courante par rapport à la valeur précédente, lorsqu'on dispose de données historiques, ou sur la différence entre la variable d'analyse et une variable auxiliaire s'il y a des données auxiliaires sûres.

PROC ERRORLOC – Localisation des erreurs (section 5)

Une fois que les règles de vérification ont été arrêtées et que les données des répondants sont disponibles, on exécute la procédure de **localisation des erreurs**. La section 5 décrit de quelle manière cette procédure applique un groupe de règles de vérification à chaque enregistrement. Lorsqu'un enregistrement ne satisfait pas aux règles, la fonction de localisation des erreurs repère les champs dans lesquels les valeurs doivent être modifiées tout en déterminant le plus petit nombre possible (pondéré) de valeurs initiales devant être modifiées pour que l'enregistrement puisse satisfaire aux règles de vérification. Cette procédure repère les champs devant faire l'objet d'une imputation, mais elle n'exécute aucune imputation.

Schiopu-Kratina et Kovar (1989) décrivent de façon détaillée de quelle manière on utilise l'algorithme de Chernikova pour solutionner le problème de localisation des erreurs.

PROC DETERMINISTIC – Imputation déterministe (section 6)

Banff offre quatre types d'imputation. L'**imputation déterministe**, qui est décrite à la section 6, sert à repérer les cas où une seule valeur possible permettra à l'enregistrement de satisfaire aux règles de vérification initiales. Dans de tels cas, l'imputation est effectuée par cette procédure.

PROC DONORIMPUTATION – Imputation par enregistrement donneur (section 7)

À la section 7, on décrit l'**imputation par enregistrement donneur**, méthode selon laquelle on impute tous les champs nécessaires d'un enregistrement en transférant les valeurs correspondantes provenant de l'enregistrement voisin le plus rapproché. Les enregistrements imputés au moyen de cette méthode sont assurés de satisfaire aux règles de vérification post-imputation spécifiées par l'utilisateur, lesquelles règles peuvent ou non être identiques aux règles initiales. Nous décrivons à la sous-section 7.1 (« **Préparer l'imputation par enregistrement donneur** ») les paramètres et les critères qui doivent être spécifiés pour obtenir les résultats d'imputation désirés. L'enregistrement voisin le plus rapproché est déterminé à partir de champs pouvant être choisis tant par le système que par l'utilisateur. On décrit ce processus à la sous-section 7.2 (« **Trouver les champs d'appariement** »). Afin de supprimer l'effet d'échelle, tous les champs d'appariement sont transformés de façon à ce que leurs valeurs s'inscrivent dans l'intervalle (0,1), comme il est décrit à la sous-section 7.3 (« **Transformer les champs d'appariement** »). Nous décrivons à la sous-section 7.4 (« **Exécuter l'imputation par enregistrement donneur** ») la méthode utilisée par Banff pour chercher l'enregistrement voisin le plus rapproché.

PROC ESTIMATOR – Imputation par estimateur (section 8)

La procédure d'**imputation par estimateur**, décrite à la section 8, permet d'imputer des champs individuels à l'aide d'une gamme d'estimateurs. Ces estimateurs peuvent être des expressions mathématiques ou provenir de modèles de régression linéaire. L'utilisateur peut définir ses propres estimateurs ou choisir parmi les 20 estimateurs prédefinis disponibles dans Banff. Plusieurs estimateurs peuvent être spécifiés et appliqués séquentiellement de manière à ce que l'on dispose de méthodes de réserve lorsqu'il n'est pas possible d'utiliser l'algorithme privilégié.

PROC PRORATE – Ajustement au prorata (section 9)

À la section 9, nous décrivons la procédure d'ajustement au prorata, par lequel on peut apporter des modifications aux composantes d'une somme afin de rendre la somme égale à un total fixe. Le processus prend en considération des règles de vérification, certains poids de champs et les codes d'état des champs, tous fournis au système par l'utilisateur. Un mécanisme d'arrondissement s'assure que toutes les données produites par le prorata aient le nombre de décimales spécifié par l'utilisateur.

PROC MASSIMPUTATION – Imputation massive (section 10)

À la section 10, nous décrivons la procédure d'imputation massive. Pour des motifs opérationnels, l'information détaillée est obtenue, dans certaines enquêtes, uniquement pour un sous-échantillon (ou échantillon de deuxième phase) d'unités sélectionnées de façon aléatoire à partir d'un échantillon de première phase, plus important. L'estimation classique basée sur le sous-échantillon nécessite le calcul de poids de sous-échantillonnage. Le calcul de ces poids peut être fort complexe. La procédure d'imputation massive applique une autre technique, qui crée un fichier rectangulaire complet pour toutes les unités de l'échantillon de première phase, en imputant par enregistrement donneur l'information manquante pour les unités non échantillonnées, après que la vérification et l'imputation pour les unités de l'échantillon de la deuxième phase ont été effectuées.

Mise à jour du présent document

Les principales fonctions de Banff ont été élaborées et elles peuvent maintenant être appliquées dans un environnement de production. Toutefois, chaque fois des modifications et des corrections sont apportées au système, les sections correspondantes de ce document seront mises à jour. On peut obtenir les versions plus récentes de n'importe quelle section en s'adressant à :

Section des systèmes généralisés
Division de la coopération internationale et méthodes statistiques institutionnelles
Statistique Canada

Les questions ou les remarques concernant le présent document ou tout aspect de Banff sont aussi les bienvenues. N'hésitez pas à contacter l'équipe du soutien de Banff si vous avez besoin d'assistance : statcan.banff-banff.statcan@canada.ca.

2. PROC VERIFYEDITS - SPÉCIFICATION ET ANALYSE DES RÈGLES DE VÉRIFICATION

2.1 SPÉCIFICATION DES RÈGLES DE VÉRIFICATION

Objet

Dans la procédure Verifyedits, l'utilisateur spécifie les conditions, ou règles de vérification, qui déterminent si un enregistrement est acceptable. Il peut spécifier les règles sous une forme qui définit soit une condition « d'acceptation » (PASS), soit une condition de « rejet » (FAIL), ou il peut combiner les deux méthodes. Il attribue ensuite chacune des règles à un ou plusieurs groupes de règles de vérification qui sont analysés dans des procédures ultérieurs et utilisés aux fins du traitement dans toute la suite de Banff.

Les règles de vérification spécifiées dans Banff

Banff vérifie des données qui sont numériques et continues. L'utilisateur spécifie les règles de vérification auxquelles doivent satisfaire les réponses contenues dans les champs de chaque enregistrement. Ces règles doivent consister en des égalités ou des inégalités linéaires de la forme suivante:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b \quad \text{ou}$$
$$a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$$

où (x_1, \dots, x_n) sont les n réponses fournies à l'enquête par une unité échantillonnée, et a_1 à a_n et b sont des constantes précisées par l'utilisateur. Quand l'utilisateur spécifie l'option « acceptnegative », règles de la forme $x_i \leq 0$ sont automatiquement ajoutées au groupe de règles spécifiées par l'utilisateur pour chaque variable qui fait l'objet d'une vérification. En conséquence, les données qui sont négatives ou manquantes ne satisfont jamais à au moins une de ces règles de vérification fondées sur la positivité des valeurs. L'utilisateur doit indiquer si chaque règle décrit une condition d'acceptation ou de rejet (PASS ou FAIL), mais il n'indique pas la mesure à prendre lorsque l'enregistrement ne satisfait pas à la règle. Voici des exemples de règles de vérification telles qu'elles seraient spécifiées par l'utilisateur.

PASS: ALIMENTS+ BOISSONS = VENTES_TOTALES

PASS: LITRES_LAIT $\geq 15 * VACHES_LAITIÈRES$

FAIL: $x_1 + x_2 + x_3 < 10$

Selon la première règle, la somme des valeurs des variables ALIMENTS et BOISSONS doit être égale à la valeur de la variable VENTES_TOTALES dans chaque enregistrement vérifié. Selon la deuxième règle, la valeur de la variable LITRES_LAIT doit être supérieure ou égale à 15 fois celle de la variable VACHES_LAITIÈRES. Pour ce qui est de la troisième règle de vérification, elle provoquerait le rejet d'un enregistrement si la somme de x_1, x_2, x_3 était inférieure à 10.

La forme canonique

Dans Banff, toutes les règles de vérification sont stockées en mémoire interne sous forme canonique, c'est-à-dire qu'elles définissent une condition d'acceptation, toutes les variables étant classées par ordre alphabétique et apparaissant à la gauche d'un opérateur « = » ou « \leq ». L'utilisateur peut entrer des règles sous n'importe quelle forme, mais le système ne fonctionne qu'avec des règles sous forme canonique et imprime les règles de vérification sous cette forme chaque fois que les résultats produits par une procédure comprennent une liste des règles de vérification. Banff renferme une fonction qui convertit les règles de vérification sous une forme canonique, fonction qui doit être exécuté chaque fois que des règles sont introduites ou modifiées. Voici des exemples de règles de vérification valides et invalides ainsi que de conversion de ces règles sous forme canonique.

RÈGLES INITIALES			FORME CANONIQUE		
PASS:	$A > B + 3$	(1)	PASS:	$-A + B \leq -3$	(1)
PASS:	$C = D$	(2)	PASS:	$C - D = 0$	(2)
PASS:	$Z < A$	(3)	PASS:	$-A + Z \leq 0$	(3)
PASS:	$M \neq N$	(4)		INVALID	
FAIL:	$A > B + 3$	(5)	PASS:	$A - B \leq 3$	(5)
FAIL:	$C = D$	(6)		INVALID	
FAIL:	$Z \leq A$	(7)	PASS:	$A - Z \leq 0$	(7)
FAIL:	$N \neq M$	(8)	PASS:	$-M + N = 0$	(8)

Il faut noter que la forme canonique ne permet pas que les règles définissant une condition d'acceptation prenne la forme d'une inégalité stricte et que les signes d'inégalité spécifiés par l'utilisateur sont automatiquement remplacés par le signe « \leq ». Ainsi, dans la règle (3), le signe « $<$ » inscrit par l'utilisateur a été remplacé par « \leq ». Dans le cas des règles définissant une condition de rejet, le signe « \leq » est remplacé par le signe « $<$ ». Ainsi, la forme canonique de la règle de vérification (7) est équivalente à FAIL: $Z < A$, même si la règle (7) a été spécifiée sous la forme FAIL: $Z \leq A$. Il faut aussi noter que l'opérateur « \neq » n'est pas un opérateur relationnel valide pour une condition d'acceptation et que l'opérateur « $=$ » n'est pas valide dans une condition de rejet. Ces restrictions s'expliquent du fait que chaque ensemble de règles de vérification doit décrire une région convexe qui contient sa propre frontière. Cette question est étudiée de façon plus détaillée à la sous-section 2.2.

Les règles de vérification non linéaires

Les règles de vérification doivent être linéaires en raison des techniques de programmation linéaire qui sont utilisées dans plusieurs procédures de Banff. Toutefois, il est possible de linéariser certains types de règles non linéaires, comme l'illustrent les deux exemples qui suivent.

$x_1 x_2 = x_3$ Il suffit de créer de nouvelles variables, $y_I = \log x_I$ pour $I = 1, 2, 3$ et la règle devient $y_1 + y_2 = y_3$. Les variables x_1 , x_2 et x_3 doivent être remplacées par y_1 , y_2 et y_3 , sinon l'imputation de l'une quelconque des variables x_1 , x_2 , x_3 , y_1 , y_2 ou y_3 pourrait entraîner l'introduction d'incohérences. Toutes les règles de vérification qui comprenaient précédemment les variables x_1 , x_2 ou x_3 doivent être remplacées par des règles comportant les variables y_1 , y_2 et y_3 . Cette transformation peut se révéler impossible si la règle en question comprend une autre variable, comme x_4 .

*si **EMP > 0** alors **SALAIRES > 0*** On peut reformuler cette règle de vérification conditionnelle pour l'exprimer sous la forme $\text{SALAIRES} \geq .000001 \text{ EMP}$, qui définit une condition légèrement plus rigoureuse que la condition initiale. Cette règle garantit que si la valeur de la variable **EMP** est supérieure à zéro, celle de la variable **SALAIRES** doit l'être aussi. Si la valeur de la variable **EMP** est égale à zéro, la valeur de la variable **SALAIRES** n'est assujettie à aucune condition. Une autre possibilité consisterait à regrouper les enregistrements dans lesquels la valeur de **EMP** est égale à zéro et à les traiter séparément de ceux dans lesquels la valeur de **EMP** est supérieure à zéro.

Bien qu'il soit possible de linéariser de nombreuses règles de vérification non linéaires, l'utilisateur qui envisage de traiter les données d'une enquête à l'aide de Banff ne devrait pas se contenter de transformer toutes les règles existantes en règles linéaires. Il devrait plutôt profiter de l'occasion pour examiner les rapports qui existent, ou qui devraient exister, entre les variables d'enquête, et établir les règles de vérification à partir de ces données et d'autres renseignements.

Règles de vérification pour le traitement des valeurs négatives

La formulation des règles de vérification pour le traitement des valeurs négatives peut poser les défis qui peuvent produire les résultats imprévus si l'utilisateur ne les anticipe pas en avance. Cette situation demande une attention spéciale. Pour plus d'information et des exemples, voyez le document « Spécification des règles de vérification avec des données négatives dans Banff » (Équipe de soutien de Banff, 2006).

Les groupes de règles de vérification et les groupes de données

Dans le SGVI, le module de localisation des erreurs qui sert à repérer les champs devant faire l'objet d'une imputation peut traiter jusqu'à près de 40 variables à la fois. Dans le cas des enquêtes de grande envergure, il faut normalement séparer les variables en groupes logiques ayant approximativement cette taille. L'ensemble de variables et les règles de vérification qui définissent les conditions auxquelles les variables doivent satisfaire forment un **groupe de règles de vérification**. Dans Banff, cette limitation du nombre de variables n'existe pas. Cependant, il est toujours utile d'organiser les règles dans un groupe pour simplifier et pour obtenir plus d'efficacité. Puis, l'utilisateur peut appliquer simplement un groupe des règles pour chaque exécution de Proc Errorloc.

Ces règles sont ensuite appliquées ensemble à chaque enregistrement d'un groupe. L'application simultanée des règles de vérification, qui constitue une des caractéristiques fondamentales de Banff, représente un atout important pour l'utilisateur, car elle permet d'éviter les incohérences susceptibles d'être introduites par l'application consécutive des règles de vérification. De telles incohérences se produisent lorsqu'on modifie un enregistrement afin qu'il satisfasse à une règle et que l'enregistrement ainsi « corrigé » ne satisfait pas à une règle subséquente et est de nouveau modifié d'une façon telle qu'il ne satisfait plus à la première règle. Cette situation ne peut survenir lorsque toutes les règles de vérification sont appliquées simultanément.

Par **groupe de données**, on entend un ensemble d'enregistrements auquel un groupe de règles de vérification s'applique. Les groupes de données peuvent être définis selon les strates d'échantillonnage, les limites provinciales, la classification des activités économiques, ou de n'importe quelle autre manière. Dans Banff, on peut traiter plus qu'un groupe de données dans une exécution d'une procédure SAS de Banff en utilisant l'énoncé BY.

Souvent, les règles de vérification varient légèrement d'une région géographique ou d'une branche d'activité à l'autre et des groupes de règles de vérification distincts doivent être créés chaque fois qu'on a besoin d'un ensemble de règles de vérification différent. Ainsi, supposons qu'une enquête a permis de recueillir des données se rapportant à six variables, x_1, x_2, \dots, x_6 , auprès de répondants appartenant à plusieurs branches d'activité. On pourrait alors créer des groupes de règles de vérification semblables à ceux qui suivent.

Groupe de règles de vérification :	Groupe de règles de vérification :	Groupe de règles de vérification :
SERVICES	VENTE AU DÉTAIL	CONSTRUCTION
$x_1 + x_2 = x_3$ (1)	$x_1 + x_2 = x_3$ (1)	$x_1 + x_2 = x_3$ (1) ...
$x_4 + x_5 = x_6$ (2)	$x_4 + x_5 = x_6$ (2)	$x_4 + x_5 = x_6$ (2) ...
$x_3 \geq 75x_6$ (3)	$x_3 \geq 80x_6$ (4)	$x_3 \geq 100x_6$ (5) ...

Comme les règles (1) et (2) sont utilisées pour toutes les branches d'activité, on les attribue à tous les groupes de règles de vérification. En revanche, la valeur de la variable x_3 doit être égale ou supérieure au produit par x_6 d'une constante dont la valeur varie selon la branche d'activité. Par conséquent, l'utilisateur définit séparément les règles (3), (4), (5), etc., puis il attribue l'une d'elles à chaque groupe de règles de vérification. Enfin, il crée des groupes de données individuels avec les enregistrements relatifs à chaque branche d'activité ou peut-être en répartissant les enregistrements selon la branche d'activité et selon la région géographique.

Il n'est pas nécessaire que les groupes de données soient les mêmes pour chaque groupe de règles de vérification, et vice versa. Prenons par exemple une enquête ayant permis de recueillir auprès de répondants de quatre régions géographiques des données sur diverses variables se rapportant aux terres, au bétail et aux cultures. Les variables étant trop nombreuses pour être vérifiées ensemble dans un seul groupe, elles sont réparties dans trois groupes différents (terres, bétail et cultures).

	Variables se rapportant aux terres	Variables se rapportant au bétail	Variables se rapportant aux cultures
	$x_1 x_2 \dots x_{20}$	$x_{21} x_{22} \dots x_{60}$	$x_{61} x_{62} \dots x_{90}$
Groupe de données de la région 1	Groupe de règles de vérification : Terre_1	Groupe de règles de vérification : Bétail_1	Groupe de règles de vérification : Cultures_1234
Groupe de données de la région 2	Groupe de règles de vérification : Terre_2	Groupe de règles de vérification :	
Groupe de données de la région 3	Groupe de règles de vérification : Terre_3	Bétail_234	
Groupe de données de la région 4	Groupe de règles de vérification : Terre_4		

Figure 2.1 Exemple de groupes de données et de groupes de règles de vérification dans une enquête sur l'agriculture

La figure 2.1 illustre une des façons dont les groupes de données et les groupes de règles de vérification peuvent être structurés. Comme les règles de vérification s'appliquant aux variables relatives aux terres sont légèrement différentes selon la région géographique, quatre groupes de règles de vérification et quatre groupes de données ont été définis pour ces variables. Un

ensemble de règles distinct a été défini pour les variables relatives au bétail des enregistrements de la région 1, tandis que les variables relatives au bétail des enregistrements des autres régions sont visées par un autre ensemble de règles de vérification, le groupe Bétail_234, et qu'un groupe de données contenant les enregistrements des régions 2, 3 et 4 a été défini. Les variables relatives aux cultures étant visées par les mêmes règles de vérification pour toutes les régions, un autre groupe de données est défini pour tous les enregistrements et le groupe de règles de vérification Cultures_1234 est utilisé pour vérifier les variables se rapportant aux cultures dans tous ces enregistrements.

Une fois qu'un groupe de règles de vérification a été créé, l'utilisateur dispose de plusieurs outils pour s'assurer que ces règles représentent de façon exacte les conditions auxquelles les données doivent être assujetties. Ces fonctions sont décrites dans les trois sous-sections suivantes :

- 2.2 Vérification des règles
- 2.3 Génération des points extrémaux
- 2.4 Génération des règles de vérification implicites

2.2 VÉRIFICATION DES RÈGLES

Objet

Cette fonction dans la procédure Verifyedits sert à vérifier si les règles faisant partie d'un groupe de règles de vérification sont cohérentes les unes avec les autres et, le cas échéant, à repérer les règles redondantes, les variables déterministes ou les égalités cachées. Une fois ces caractéristiques décelées, on peut déterminer l'ensemble de règles de vérification minimal. Aucune donnée fournie par les répondants n'est utilisée dans cette procédure; ce sont les règles de vérification elles-mêmes qui sont analysées.

Définitions

Un groupe de règles de vérification comprenant n variables définit une région, appelée **région d'acceptation**, dans l'espace à n dimensions. Lorsque les valeurs d'enquête initiales sont substituées aux variables dans les égalités ou inégalités qui composent le groupe de règles de vérification, les enregistrements qui satisfont à toutes les règles se situent dans la région d'acceptation, tandis que les enregistrements qui ne satisfont pas à toutes les règles se situent à l'extérieur de cette région. La région d'acceptation utilisée par Banff doit être convexe et doit inclure sa frontière. Lorsqu'une condition d'acceptation est spécifiée par l'opérateur « $<$ », ce dernier est automatiquement remplacé par « \leq » dans la forme canonique de manière à ce que la frontière soit incluse dans la région d'acceptation. Une condition d'acceptation ne peut pas être exprimée sous la forme d'une relation « \neq » parce que la région d'acceptation résultante serait formée de deux régions distinctes situées de chaque côté de la relation d'égalité et qu'elle ne serait pas convexe.

Une région d'acceptation peut être décrite par l'un quelconque d'un nombre infini d'ensembles de règles de vérification différents. Un **ensemble minimal de règles de vérification** renferme le plus petit nombre de règles nécessaires pour définir une certaine région d'acceptation. La procédure de vérification des règles fournit à l'utilisateur les renseignements nécessaires pour créer un ensemble de règles minimal pour la région d'acceptation désirée. Cet ensemble minimal devrait être utilisé dans la poursuite du traitement afin d'améliorer l'efficacité de ce dernier.

La fonction de vérification des règles fournit aussi des renseignements qui devraient permettre à l'utilisateur de mieux comprendre la région d'acceptation définie par un groupe de règles de vérification. Dans Banff, la détermination des champs devant faire l'objet d'une imputation, tout comme le succès de certaines méthodes d'imputation, sont fonction du groupe de règles de vérification; il est donc très important que les règles de vérification représentent de façon exacte les conditions auxquelles l'utilisateur désire assujettir les données.

La présente fonction sert à vérifier si les règles faisant partie d'un groupe de règles de vérification sont cohérentes et à repérer les règles redondantes, les variables déterministes et les égalités cachées. Ces termes sont définis et illustrés dans la présente sous-section, mais nous n'y exposons pas de façon détaillée les méthodes réellement utilisées dans Banff. Bon nombre de ces méthodes exigent qu'on maximise et/ou qu'on minimise une règle de vérification donnée, eu égard aux autres contraintes. Cette opération est effectuée à l'aide d'une version révisée de la méthode du simplexe (inverse de la forme du produit). La méthode de Givens est utilisée pour convertir une matrice sous une forme triangulaire lorsqu'on vérifie s'il y a des égalités cachées. Le lecteur qui est intéressé à obtenir une description plus complète des méthodes employées dans la procédure de vérification des règles de Banff est invité à se reporter à l'ouvrage de Giles (1989).

Groupes de règles de vérification cohérents et incohérents

Un groupe cohérent de règles de vérification définit une région d'acceptation non vide. Un groupe incohérent de règles de vérification contient des règles auxquelles aucun enregistrement de données ne pourra jamais satisfaire parce que certaines de ces règles sont contradictoires. Les figures 2.2 et 2.3 illustrent ces deux types de groupe.

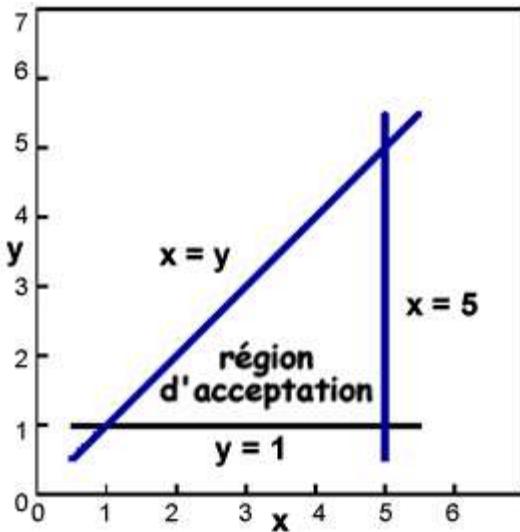


Fig. 2.2 Groupe cohérent de règles de vérification
 $x \geq y$ (1)
 $x \leq 5$ (2)
 $y \geq 1$ (3)

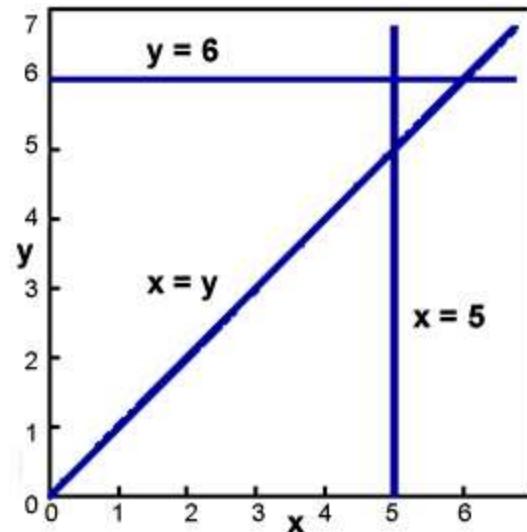


Fig. 2.3 Groupe incohérent de règles de vérification
 $x \geq y$ (1)
 $x \leq 5$ (2)
 $y \geq 6$ (4)

Le groupe de règles de vérification de la figure 2.2 est cohérent et il définit une région d'acceptation, tandis que le groupe de la figure 2.3 est incohérent. Il ne permet de définir aucune région d'acceptation parce qu'aucun point ne satisfait aux trois règles en même temps, bien qu'il existe des régions qui satisferaient à chacune des trois paires possibles de règles de vérification. Dans l'exemple ci-dessus, il est possible d'utiliser un graphique pour repérer une règle de vérification inappropriée ou incorrectement spécifiée, mais, dans une application type, le groupe de règles de vérification comprend de nombreuses variables et il n'est pas possible de vérifier la région d'acceptation à l'aide d'un graphique.

Lorsqu'il repère un groupe de règles de vérification incohérent, Banff indique à l'utilisateur un sous-ensemble des règles devant être éliminées de l'ensemble pour en assurer la cohérence.

L'utilisateur peut considérer le fait que le système lui précise un sous-ensemble de règles devant être supprimées comme une indication qu'il existe un problème, mais il se doit de garder d'en déduire qu'il suffit de supprimer ces règles et de poursuivre le traitement. Tout ensemble de règles de vérification incohérent doit être réexaminé minutieusement, parce qu'il signifie que l'utilisateur a spécifié des conditions qui sont contradictoires. La première étape consiste à vérifier si toutes les règles ont été entrées correctement, car une divergence même légère peut se traduire par une règle très différente de la règle voulue. Si toutes les règles ont été spécifiées correctement, l'utilisateur peut, s'il le veut, supprimer des règles une à une ou en combinaison afin de définir divers ensembles cohérents. L'objectif n'est pas de supprimer le moins de règles possible, mais de déterminer un ensemble de règles de vérification cohérent qui représente les conditions auxquelles l'utilisateur souhaite assujettir les données.

Règles redondantes; règles strictes mais non restrictives

La procédure de vérification des règles repère aussi les règles **redondantes**. Par règle redondante, on entend une règle qui définit une courbe ne faisant pas partie de la frontière de la région d'acceptation et, par conséquent, n'imposant aucune restriction sur les valeurs acceptables. Les règles de vérification **strictes mais non restrictives** sont des cas particuliers des règles redondantes. Ces règles définissent des courbes qui touchent la frontière de la région d'acceptation en un seul point, mais qui n'imposent aucune contrainte sur les valeurs admissibles, compte tenu des autres règles de vérification. Ni les règles redondantes ni les règles strictes mais non restrictives n'appartiennent à l'ensemble de règles de vérification minimal nécessaire pour définir une région d'acceptation. Ces règles doivent être omises dans la poursuite du traitement afin d'améliorer l'efficacité de ce dernier. On trouve un exemple de règle redondante et un exemple de règle stricte mais non restrictive aux figures 2.4 et 2.5 respectivement.

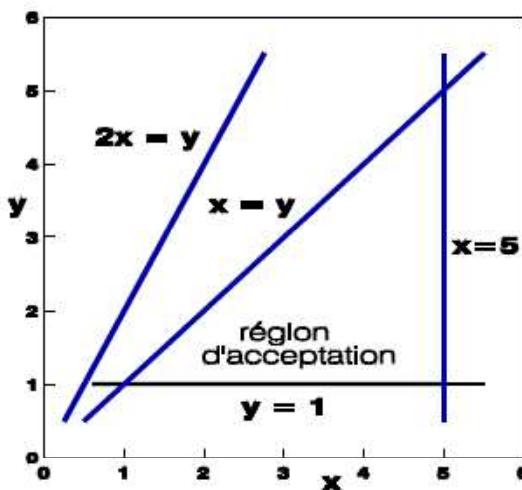


Fig. 2.4 La règle (4) est redondante

$$\begin{aligned} x \geq y & \quad (1) \\ x \leq 5 & \quad (2) \\ y \geq 1 & \quad (3) \\ 2x \geq y & \quad (4) \quad (\text{redondante}) \end{aligned}$$

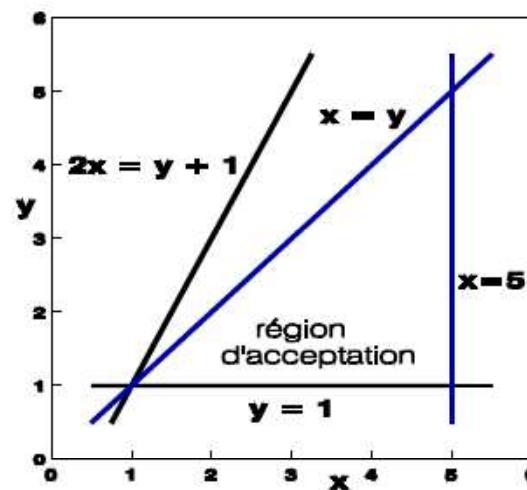


Fig. 2.5 La règle (5) est stricte mais non-restrictive

$$\begin{aligned} x \geq y & \quad (1) \\ x \leq 5 & \quad (2) \\ y \geq 1 & \quad (3) \\ 2x \geq y + 1 & \quad (5) \quad (\text{stricte mais non-restrictive}) \end{aligned}$$

Dans la figure 2.4, la règle (4) ne définit pas une courbe faisant partie de la frontière de la région d'acceptation: elle est donc redondante. Tout enregistrement qui satisfait aux règles de vérification (1), (2) et (3) satisfera automatiquement à la règle (4). Dans la figure 2.5, la règle (5) définit une courbe qui touche la région d'acceptation en un seul point, mais qui n'impose toujours pas aucune contrainte sur les valeurs acceptables. Tout enregistrement qui satisfait aux règles de vérification (1), (2) et (3) satisfera automatiquement à la règle (5). Dans les deux cas, l'utilisateur doit étudier les règles de vérification spécifiées et décider si les règles redondantes et les règles strictes doivent être éliminées ou si ce sont les autres règles de vérification qui sont trop restrictives. Dans le cas des exemples ci-dessus, l'utilisateur pourrait spécifier un nouveau groupe de règles de vérification composé des règles (1), (2) et (3), ou il pourrait choisir un nouveau groupe formé des règles (2), (3) et (4) dans le cas de la figure 2.4, et des règles (2), (3) et (5) dans celui de la figure 2.5. Dans tous les cas, le nouveau groupe serait utilisé pour tout le traitement ultérieur.

Égalités cachées

Un groupe de règles de vérification peut contenir des inégalités qui, lorsqu'on les combine avec les contraintes imposées par les autres règles, sous-entendent une égalité. Il peut arriver que ces règles de vérification, appelées **égalités cachées**, ne soient pas évidentes lorsque plusieurs variables sont en jeu. La présence d'égalités cachées indique en général que les règles sont plus restrictives que l'utilisateur ne l'avait cru. Lorsque la procédure de vérification des règles repère des égalités cachées, il faut réexaminer les règles afin de déterminer si l'égalité cachée doit être retenue ou si une ou plusieurs règles de vérification ont été incorrectement spécifiées. Prenons par exemple le groupe de règles de vérification suivant:

$$\begin{aligned} x_1 + x_2 + x_4 + x_5 &\leq 4 \quad (1) \\ x_2 + x_3 - x_4 + x_5 &\geq 2 \quad (2) \\ x_1 + x_4 &= 3 \quad (3) \\ x_3 - x_4 &= 1 \quad (4) \end{aligned}$$

Dans cet exemple, la procédure de vérification de règles déterminerait que les règles (1) et (2) forment une égalité cachée et désignerait la règle (2) comme étant une **égalité cachée redondante**. On peut voir plus clairement les égalités cachées contenues dans ce groupe de règles de vérification après avoir exécuté de simples substitutions algébriques.

Substituons la règle (3) dans la règle (1) pour obtenir : $x_2 + x_5 \leq 1$

Substituons la règle (4) dans la règle (2) pour obtenir : $x_2 + x_5 \geq 1$

Si l'utilisateur décide que les règles sont correctement spécifiées, alors les règles $x_2 + x_5 \leq 1$ et $x_2 + x_5 \geq 1$ doivent être toutes les deux vraies, ce qui bien sûr sous-entend que $x_2 + x_5 = 1$. Les règles (1) et (2) devraient donc être remplacées par $x_2 + x_5 = 1$, ou par une version révisée de la règle (1) ou (2) énoncée sous la forme d'une égalité. Dans tous les cas, le groupe formé par la nouvelle règle combinée aux règles (3) et (4) définit la même région d'acceptation que les règles initiales. Voici ces trois groupes de règles de vérification équivalents.

Groupes de règles de vérification équivalents

$$\begin{array}{lll} x_2 + x_5 = 1 & (5) & x_1 + x_2 + x_4 + x_5 = 4 \quad (1a) & x_2 + x_3 - x_4 + x_5 = 2 \quad (2a) \\ x_1 + x_4 = 3 & (3) & x_1 + x_4 = 3 \quad (3) & x_1 + x_4 = 3 \quad (3) \\ x_3 - x_4 = 1 & (4) & x_3 - x_4 = 1 \quad (4) & x_3 - x_4 = 1 \quad (4) \end{array}$$

Par contre, l'utilisateur peut décider que les règles ne sont pas correctement spécifiées et que, disons, l'opérateur « \leq » dans la règle (1) devrait être remplacé par l'opérateur « \geq ». Si ce changement est apporté, la procédure de vérification des règles déterminerait que la règle (2) est stricte mais non restrictive et ne trouverait aucune égalité cachée parmi les règles de vérification.

Limites supérieures et inférieures; variables déterministes

La procédure de vérification des règles produit aussi les **limites supérieure et inférieure** pour chaque variable. Ces limites sont les valeurs maximale et minimale qu'une variable peut prendre tout en demeurant à l'intérieur de la région d'acceptation définie par les règles de vérification. Les limites ne représentent pas les valeurs réelles des données, car aucune donnée fournie par les répondants n'est utilisée dans cette procédure. L'utilisateur doit examiner les limites et envisager d'ajouter ou de supprimer des règles si les limites ne semblent pas raisonnables. Pour illustrer notre propos, examinons le groupe de règles de vérification suivant et les limites qui seraient produites par la procédure de vérification des règles.

$$\begin{array}{ll} x_1 + x_2 + x_4 = 10 & (1) \\ x_1 + x_2 = 6 & (2) \\ x_3 + x_4 \geq 8 & (3) \\ x_4 \geq 0 & (7) \end{array} \quad \begin{array}{ll} x_1 \geq 0 & (4) \\ x_2 \geq 0 & (5) \\ x_3 \geq 0 & (6) \end{array}$$

Variable	Limite inférieure	Limite supérieure	
x_1	0	6	
x_2	0	6	
x_3	4	*****	ILLIMITÉE
x_4	4	4	DÉTERMINISTE

Tant x_1 que x_2 ont une valeur minimale de 0 et une valeur maximale de 6. La variable x_3 ne peut pas prendre une valeur valide inférieure à 4, mais il n'y a pas de limite quant à sa valeur maximale. Pour ce qui est de la variable x_4 sa limite inférieure est de 4, tout comme sa limite supérieure; la valeur de cette variable doit donc toujours être égale à 4. Cette variable est **déterministe**, c'est-à-dire qu'elle ne peut prendre qu'une seule valeur possible. Le caractère déterministe d'une variable peut être dû au fait qu'une des règles de vérification a été incorrectement spécifiée; l'utilisateur doit donc passer en revue tout le groupe de règles afin de décider si la variable désignée comme étant déterministe devrait être limitée à une seule valeur possible dans le groupe d'enregistrements en cours de traitement.

Les limites inférieures et supérieures ne tiennent pas compte du fait que les valeurs valides dans un champ d'un enregistrement particulier peuvent dépendre des valeurs dans les autres champs. Dans l'exemple ci-dessus, les règles de vérification ne permettent pas que x_1 et x_2 prennent leurs valeurs maximales ou minimales en même temps. De fait, lorsqu'une atteint son maximum, l'autre est à son minimum, comme l'indique leur relation dans la règle (2).

2.3 GÉNÉRATION DES POINTS EXTRÉMAUX

Objet

Cette fonction de la procédure Verifyedits génère tous les points extrémaux, ou sommets, de la région d'acceptation définie par un groupe de règles de vérification. Ces points représentent les enregistrements de données les plus extrêmes qui seraient acceptables: ils peuvent par conséquent permettre à l'utilisateur de mieux comprendre la forme de la région d'acceptation qui est précisée.

Description de la méthode employée

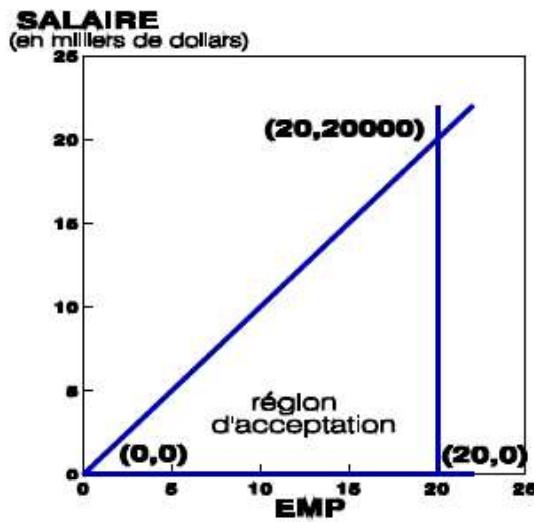
Chaque **point extrémal** est un sommet de la région d'acceptation qui peut être représenté par ses coordonnées dans l'espace à n dimensions. Sur le plan géométrique, les points extrémaux se situent à l'intersection de n règles, où n est le nombre de variables. Dans le contexte d'une enquête, les points extrémaux peuvent être considérés comme étant les valeurs les plus extrêmes que peuvent prendre les variables contenues dans un enregistrement, tout en demeurant acceptables.

On peut trouver dans l'ouvrage de Schiopu-Kratina et de Kovar (1989) une description complète de la façon d'utiliser l'algorithme de Chernikova pour générer tous les points extrémaux de la région d'acceptation. Une matrice est créée à partir des règles de vérification spécifiées par l'utilisateur et des règles de vérification implicites fondées sur la positivité des valeurs. Cette matrice est transformée au moyen d'une série d'itérations dans le cadre de chacune desquelles les colonnes sont soit conservées, soit prises en combinaison linéaire avec d'autres pour produire de nouvelles colonnes. Les itérations se poursuivent jusqu'à ce qu'on ait générés tous les points extrémaux ou jusqu'à ce qu'on ait produit tous les points extrémaux dont la cardinalité est inférieure ou égale à une limite précisée par l'utilisateur. On entend ici par **cardinalité** le nombre de coordonnées non nulles d'un point extrémal. Par exemple, le point (5, 0, 10) a une cardinalité de deux. Les points les plus faciles à interpréter sont souvent ceux qui ont de nombreuses valeurs nulles parce que, dans un sens, ces variables sont éliminées. Ainsi, l'utilisateur peut choisir de restreindre la génération des points extrémaux à ceux dont le nombre de coordonnées non nulles est inférieur ou égal à une limite précisée par l'utilisateur. La limitation de la cardinalité peut aussi permettre de réduire le temps d'exécution.

Le nombre de points extrémaux liés à un groupe de règles de vérification ne doit pas dépasser une limite supérieure théorique, mais le nombre réel de points extrémaux est d'ordinaire de beaucoup inférieur à cette limite. Souvent, tant le nombre réel que la limite théorique sont très élevés, même pour des groupes de règles de vérification de taille moyenne. Le lecteur trouvera de plus amples détails à ce sujet dans le document de Giles (1989).

Exemples de points extrémaux

Prenons par exemple les règles de vérification établies pour un groupe de petites entreprises dans un secteur d'activité donné.



RÈGLES INITIALES

$\text{EMP} \leq 20$
 $\text{SALAIRE} \leq 1000 \text{ EMP}$

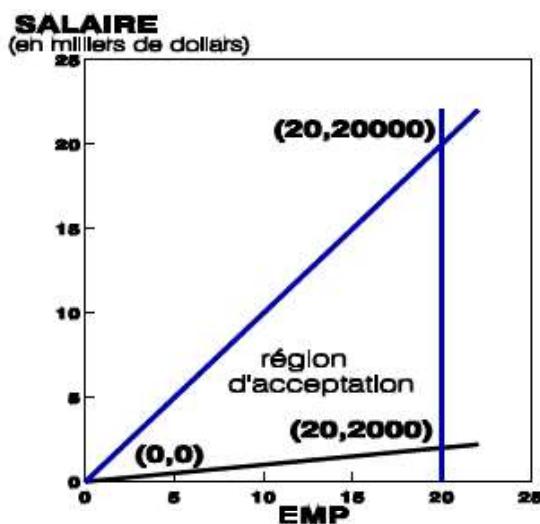
$\text{EMP} \geq 0$ (règle implicite fondée sur la positivité des valeurs)
 $\text{SALAIRE} \geq 0$ (règle implicite fondée sur la positivité des valeurs)

POINTSEXTRÉMAUX

(0 , 0) cardinalité de 0
(20 , 20000) cardinalité de 2
(20, 0) cardinalité de 1

Figure 2.6 Points extrémaux du groupe de règles de vérification initial

L'examen des trois points extrémaux indiqués à la figure 2.6 peut amener l'utilisateur à douter de la valeur du point (20,0). Une entreprise peut-elle avoir 20 employés et ne verser aucun salaire? Si l'utilisateur décide que cette règle n'est pas acceptable, il pourrait ajouter une autre règle de vérification afin d'imposer un salaire moyen minimal. Un groupe de règles de vérification révisé serait alors soumis aux fonctions de génération de la forme canonique, de vérification des règles et de génération des points extrémaux. Avec le nouveau groupe de règles illustré à la figure 2.7, le point (20,0) se trouverait à l'extérieur de la région d'acceptation et un enregistrement comportant ces valeurs serait repéré comme étant un enregistrement inacceptable dans une procédure Banff ultérieur.



RÈGLES RÉVISÉES

$\text{EMP} \leq 20$
 $\text{SALAIRE} \leq 1000 \text{ EMP}$
 $\text{SALAIRE} \geq 100 \text{ EMP}$
 $\text{EMP} \geq 0$
 $\text{SALAIRE} \geq 0$

POINTSEXTRÉMAUX

(0 , 0)
(20 , 20000)
(20 , 2000)

Figure 2.7 Points extrémaux du groupe de règles de vérification révisé

2.4 GÉNÉRATION DES RÈGLES DE VÉRIFICATION IMPLICITES

Objet

Cette fonction de la procédure Verifyedits génère les règles de vérification additionnelles qui sont sousentendues par un groupe de règles. L'utilisateur peut examiner ces règles implicites pour s'assurer que toutes les conditions sous-entendues par le groupe de règles de vérification sont acceptables.

Description de la méthode employée

Le concept des règles de vérification implicites a été introduit par Fellegi et Holt (1976). Dans Banff, une **règle de vérification implicite** est le résultat d'une combinaison linéaire de k règles dans lesquelles au moins $(k-1)$ variables ont été éliminées. Les règles implicites sont toujours redondantes et ne doivent donc jamais être ajoutées à l'ensemble de règles de vérification minimal. Elles peuvent toutefois comprendre moins de variables que les règles spécifiées par l'utilisateur et, de ce fait, mettre en lumière des conditions dont ce dernier n'avait pas conscience.

On peut trouver dans le document de Schiopu-Kratina et de Kovar (1989) une description complète de la façon d'utiliser l'algorithme de Chernikova pour générer les règles de vérification implicites. Une matrice est créée à partir des règles de vérification spécifiées par l'utilisateur et des règles fondées sur la positivité des valeurs. La transposée de cette matrice subit des transformations successives à mesure que chacune de ses lignes est traitée. À chaque itération, les colonnes sont prises en combinaison linéaire avec d'autres pour produire des colonnes additionnelles. Ces itérations se poursuivent jusqu'à ce que toutes les lignes aient été traitées ou jusqu'à ce que le nombre de règles de vérification implicites produites excède une limite précisée par l'utilisateur. L'utilisateur peut, s'il le désire, appliquer cette limite en raison du grand nombre de règles implicites qui peuvent être générées à partir d'un nombre relativement petit de règles de vérification, surtout en présence d'une règle fondée sur une égalité.

Exemples de règles de vérification implicites

Soit le groupe suivant composé de deux règles de vérification spécifiées par l'utilisateur et de trois règles fondées sur la positivité des valeurs, ajoutées par le système :

$$x_1 - 2x_2 + 3x_3 \leq 10 \quad (1)$$

$$x_1 + x_2 + x_3 \leq 5 \quad (2)$$

$-x_1 \leq 0$ (3) (règle fondée sur la positivité des valeurs)

$-x_2 \leq 0$ (4) (règle fondée sur la positivité des valeurs)

$-x_3 \leq 0$ (5) (règle fondée sur la positivité des valeurs)

Les règles (1) et (2) peuvent être combinées pour produire la règle implicite (6), laquelle peut elle-même être combinée avec la règle (3) pour produire la règle (7).

$$\begin{array}{l} x_1 - 2x_2 + 3x_3 \leq 10 \quad (1) \\ 2x_1 + 2x_2 + 2x_3 \leq 10 \quad (2) \times 2 \\ \hline 3x_1 + 5x_3 \leq 20 \quad (6) \end{array} \qquad \begin{array}{l} 3x_1 + 5x_3 \leq 20 \quad (6) \\ -3x_1 \leq 0 \quad (3) \times 3 \\ \hline 5x_3 \leq 20 \quad (7) \end{array}$$

Il est courant d'avoir un très grand nombre de règles de vérification implicites. Ce petit groupe de règles composé de deux règles fondées sur une inégalité et de trois règles fondées sur la positivité des valeurs produit neuf règles implicites. On trouve ci-après la liste complète des règles implicites comme elle apparaîtrait dans les résultats produits par la procédure. On notera que le coefficient principal de la première variable de chaque équation est toujours soit +1, soit -1; par exemple, la règle (7) apparaît sous la forme $x_3 \leq 4$ plutôt que sous la forme $5x_3 \leq 20$ comme ci-dessus.

Liste des règles implicites

$$\begin{aligned}
 & x_2 + x_3 \leq 5 \\
 & -x_2 + 1.5x_3 \leq 5 \\
 & x_1 + 1.66667x_3 \leq 6.66667 \quad (6) \\
 & x_1 + x_3 \leq 5 \\
 & x_3 \leq 4 \quad (7) \\
 & x_1 + x_2 \leq 5 \\
 & x_1 - 2x_2 \leq 10 \\
 & x_2 \leq 5 \\
 & x_1 \leq 5
 \end{aligned}$$

En examinant les règles implicites, l'utilisateur peut découvrir qu'il y a entre les variables des rapports qui sont inacceptables. En pareil cas, il doit examiner et modifier les règles initiales, puis soumettre le nouvel ensemble de règles de vérification à la procédure Verifyedits.

3. PROC EDITSTATS – STATISTIQUES SOMMAIRES SUR LES RÈGLES DE VÉRIFICATION

Objet

Cette procédure applique un groupe de règles de vérification aux enregistrements contenant les données fournies par les répondants et détermine si chaque enregistrement satisfait ou non ou a des valeurs manquantes à chaque règle. Les codes d'état créés pendant ce processus sont utilisés uniquement dans cette procédure et ne sont pas transmis aux autres procédures. La procédure génère cinq tableaux faisant la récapitulation des codes d'état attribués, lesquels tableaux peuvent servir à améliorer le groupe de règles de vérification, à estimer les ressources nécessaires pour la mise en oeuvre des procédures ultérieurs ou à évaluer les effets de l'imputation. Lorsqu'un enregistrement ne satisfait pas à une règle, le système n'essaie pas d'indiquer quelles modifications doivent être apportées à l'enregistrement pour qu'il satisfasse à ladite règle.

Description de la méthode employée

Pour un groupe de règles de vérification composé de m règles fondées sur la positivité des valeurs et de n règles spécifiées par l'utilisateur, sans règles redondantes, un grand total de $m + n + 1$ codes d'état sont attribués à chaque enregistrement. À ce stade-ci, les règles redondantes devraient avoir été supprimées du groupe de règles de vérification; toutefois, si elles sont encore incluses, le nombre de codes d'état à générer s'en trouvera réduit puisque les règles redondantes n'apparaissent pas dans les tableaux. L'opération suivante est exécutée indépendamment pour chaque enregistrement de données.

- Un code d'état est attribué à l'enregistrement de données pour chaque règle de vérification, y compris les règles fondées sur la positivité des valeurs. Il y a $m + n$ de ces codes au total. L'état peut être :
 - PASS, si l'enregistrement satisfait à la règle;
 - MISS, si une ou plusieurs valeurs visées par la règle sont manquantes; ou
 - FAIL, si l'enregistrement ne satisfait pas à la règle à cause d'une ou plusieurs valeurs non manquantes.
- Un état général de l'enregistrement est produit à partir des $m + n$ codes d'état qui lui ont été attribués en fonction des résultats de l'application de chacune des règles de vérification. Cet état général peut être :
 - PASS, si le code d'état attribué pour chaque règle est PASS, c'est-à-dire si l'enregistrement a satisfait à toutes les règles du groupe;
 - MISS, si le code d'état attribué pour une ou plusieurs règles est MISS et si aucun code d'état FAIL n'a été attribué, c'est-à-dire si pour chaque règle à laquelle l'enregistrement n'a pas satisfait, des valeurs étaient manquantes; ou
 - FAIL, si le code d'état attribué pour une ou plusieurs règles est FAIL, c'est-à-dire si l'enregistrement n'a pas satisfait à certaines règles à cause de valeurs non manquantes et éventuellement à cause de valeurs manquantes.

Exemple de création de codes d'état pour chaque règle

Prenons par exemple le groupe de règles de vérification et les enregistrements de données suivants.

Règles fondées sur la positivité des valeurs	Règles spécifiées par l'utilisateur
$x_1 \geq 0$ (1)	$x_1 + 1 \geq x_2$ (4)
$x_2 \geq 0$ (2)	$x_1 \leq 5$ (5)
$x_3 \geq 0$ (3)	$x_2 \geq x_3$ (6)
	$x_1 + x_2 + x_3 \leq 9$ (7)

Il y a trois règles fondées sur la positivité des valeurs et quatre règles spécifiées par l'utilisateur; il faut donc attribuer huit codes d'état ($3 + 4 + 1$) à chaque enregistrement. Ces codes apparaissent sous les rubriques « État pour chaque règle » ((1) à (7)) et « État général » dans le tableau qui suit.

	Enregistrements de données			État pour chaque règle							État général
	x_1	x_2	x_3	(1)	(2)	(3)	(4)	(5)	(6)	(7)	
enregistrement 1	4	3	2	P	P	P	P	P	P	P	P
enregistrement 2	4	3	manquante	P	P	M	P	P	M	M	M
enregistrement 3	6	3	2	P	P	P	P	F	P	F	F
enregistrement 4	6	3	manquante	P	P	M	P	F	M	M	F

Comme l'enregistrement 1 satisfait à toutes les règles de vérification, un code d'état PASS lui a été attribué pour chacune des sept règles et pour l'état général. L'enregistrement 2 satisfait à toutes les règles, sauf celles s'appliquant à la variable x_3 , qui est manquante. Ainsi, on a attribué le code d'état PASS dans le cas des règles (1), (2), (4) et (5), et le code d'état MISS dans le cas des règles (3), (6) et (7). Le code d'état général de l'enregistrement 2 est aussi MISS. L'enregistrement 3 ne comporte aucune valeur manquante, mais il ne satisfait pas aux règles (5) et (7). Comme on a attribué un code d'état PASS pour toutes les règles à l'exception des règles (5) et (7), pour lesquelles on a attribué un code d'état FAIL, l'état général de l'enregistrement 3 est FAIL. L'enregistrement 4 ne satisfait pas à la règle (5) à cause de valeurs non manquantes « incorrectes », et il ne satisfait pas à d'autres règles en raison de valeurs manquantes. On a attribué le code d'état PASS dans le cas des règles (1), (2) et (4), le code d'état FAIL en ce qui concerne la règle (5) et le code d'état MISS pour ce qui est des règles (3), (6) et (7). Le code d'état général de l'enregistrement 4 est FAIL parce que ce code d'état a la priorité sur le code d'état MISS lorsqu'on détermine le code d'état général de l'enregistrement.

Exemples de tableaux

On trouve ci-après une brève description des cinq tableaux de statistiques sommaires sur les règles de vérification, ainsi que les tableaux qui seraient produits pour les règles de vérification et les enregistrements de données utilisés dans l'exemple précédent de création de codes d'état. L'utilisateur ne doit pas oublier que les chiffres contenus dans les tableaux 1-1 et 1-2 ont été calculés à partir des codes d'état attribués pour chaque règle, tandis que les chiffres qu'on trouve dans le tableau 1-3 ont été calculés à partir des codes d'état général attribués aux enregistrements. Par ailleurs, le tableau 2-1 fait état de chiffres calculés à partir des codes d'état pour chaque règles de vérification attribués à chaque champ de l'enregistrement suivant les règles décrites ci-après, alors que le tableau 2-2 fait état de chiffres calculés à partir des codes d'état général qui ont été attribués à chaque champ de l'enregistrement, suivant les règles décrites ci-après.

TABLEAU 1-1

Totaux des observations avec l'état passed (succès), missed (manque) ou failed (échec) pour chaque règle

EDITID	OBS_PASSED	OBS_MISSED	OBS_FAILED
POSITIVITY EDIT x ₁	4	0	0
POSITIVITY EDIT x ₂	4	0	0
POSITIVITY EDIT x ₃	2	2	0
EDIT (4)	4	0	0
EDIT (5)	2	0	2
EDIT (6)	2	2	0
EDIT (7)	1	2	1

Le tableau 1-1 indique, pour chaque règle de vérification, le nombre d'enregistrements qui ont satisfait à la règle, le nombre de ceux qui n'y ont pas satisfait et le nombre de ceux qui comportaient des valeurs manquantes. Les chiffres qu'il contient ont été établis à partir des codes d'état PASS, MISS ou FAIL qui ont été attribués pour chaque règle. Comme le système attribue à chaque enregistrement le code d'état PASS, MISS ou FAIL pour chacune des règles, le total de chaque ligne devrait être égal au nombre total d'enregistrements.

Ainsi, les quatre enregistrements ont satisfait à la règle (1), qui stipule que la variable x1 doit être positive; le chiffre 4 figure donc dans la colonne sous la rubrique « OBS_PASSED » et le chiffre 0, dans les colonnes sous les rubriques « OBS_MISSED » et « OBS_FAILED ». Par ailleurs, seul l'enregistrement 1 a satisfait à la règle (7), alors que les enregistrements 2 et 4 se sont vus attribuer le code d'état MISS et que l'enregistrement 3 n'a pas satisfait à cette règle. On trouve donc à la dernière ligne du tableau 1-1 les chiffres 1, 2 et 1 sous les rubriques « OBS_PASSED », « OBS_MISSED » et « OBS_FAILED » respectivement.

À partir du tableau 1-1, l'utilisateur peut déterminer si les enregistrements ont tendance à être rejetés ou à comporter des valeurs manquantes plus souvent pour certaines règles que pour d'autres. Si c'est le cas, il peut arriver que les règles soient trop restrictives ou que la qualité des données laisse à désirer.

TABLEAU 1-2

Distribution des observations qui ont passé avec succès (passed), un manque (missed) ou un échec (failed) un nombre K de règles

NUMBER OF EDITS (K_EDITS)	OBS_PASSED	OBS_MISSED	OBS_FAILED
0	0	2	2
1	0	0	1
2	0	0	1
3	1	2	0
4	1	0	0
5	1	0	0
6	0	0	0
7	1	0	0
TOTAL RECORDS	4	4	4

Le tableau 1-2 donne la répartition des enregistrements selon le nombre de fois où chaque code d'état leur a été attribué par suite de l'application d'une règle de vérification. Les chiffres qu'il renferme, qui ont été établis à partir des codes d'état qui ont été attribués selon chaque règle, indiquent le nombre d'enregistrements pour lesquels chaque code d'état a été attribué zéro fois, une fois, deux fois, etc. Chaque enregistrement est compté une fois dans chaque colonne (à la ligne correspondant au nombre de règles pour lesquelles on lui a attribué un code d'état donné) de sorte que le total de chaque colonne est égal au nombre d'enregistrements.

Dans cet exemple, la première ligne du tableau 1-2 indique qu'aucun enregistrement n'a pas reçu le code d'état PASS, tandis que deux enregistrements ne se sont vus attribuer le code d'état MISS et deux autres, le code d'état FAIL. Si l'on descend à la ligne où le chiffre « 3 » figure dans la colonne de gauche, on peut voir qu'un enregistrement a reçu le code d'état PASS trois fois, que deux enregistrements se sont vus attribuer le code MISS trois fois et qu'aucun enregistrement ne s'est vu attribuer le code FAIL trois fois. La ligne commençant par le chiffre « 7 » indique qu'un enregistrement a reçu le code PASS sept fois et qu'aucun enregistrement ne s'est vu attribuer le code d'état MISS pour les sept règles, de même pour le code d'état FAIL.

À partir du tableau 1-2, l'utilisateur peut déterminer si les données analysées font partie d'un nombre modéré d'enregistrements qui satisfont à toutes les règles de vérification et d'un nombre modéré d'enregistrements qui ne satisfont pas à quelques-unes de ces règles, ou si elles font partie d'un grand nombre d'enregistrements qui satisfont à toutes les règles et d'un nombre beaucoup plus petit d'enregistrements qui ne satisfont pas à la plupart ou à la totalité des règles.

TABLEAU 1-3

Totaux globaux des observations avec l'état passed (succès), missed (manque) ou failed (échec)

OBS_PASSED	OBS_MISSED	OBS FAILED	OBS_TOTAL
1	1	2	4

Le tableau 1-3 indique le nombre d'enregistrements dont l'état général est PASS, MISS ou FAIL. Chaque enregistrement n'étant compté qu'une seule fois, le total des chiffres figurant à la seule ligne de ce tableau devrait donc être égal au nombre total d'enregistrements.

Dans l'exemple utilisé dans la présente section, un enregistrement s'était vu attribuer le code d'état général PASS, un autre le code MISS et les deux autres le code FAIL. Il faut noter que les enregistrements dont le code d'état général est FAIL ont pu se voir attribuer le code d'état MISS pour une ou plusieurs règles de vérification ainsi que le code FAIL pour au moins une règle.

La somme des cases OBS_MISSED et OBS_FAILED du tableau 1-3 donne un aperçu du nombre de rejets auquel il faudra s'attendre au moment de l'exécution du la procédure de localisation des erreurs.

TABLEAU 2-1

Totaux d'application des règles avec l'état pass (succès), miss (manque) ou fail (échec) impliquant chaque champ

FIELDID	EDIT_APPLIC_PASSED	EDIT_APPLIC_MISSED	EDIT_APPLIC_FAILED	EDIT_APPLIC_NOT INVOLVED	EDITS_INVOLVED
X 1	11	2	3	12	4
X 2	11	4	1	12	4
X 3	5	6	1	16	3

Le tableau 2-1 indique le nombre de fois où chaque variable a été visée par une règle pour laquelle le système a attribué le code d'état PASS, MISS ou FAIL. Aux fins de cette totalisation, le système attribue à chaque variable visée par la règle les codes d'état PASS, MISS et FAIL qui ont été générés pour un enregistrement et une règle en particulier, tandis qu'il attribue un code NOT INVOLVED aux variables non visées par la règle. Il fait ensuite le total du nombre de fois où chaque code a été attribué à chaque variable. Pour une ligne déterminée du tableau (c'est-à-dire pour une variable donnée), chaque enregistrement est compté avec chaque règle, de sorte que le total de toutes les cases sauf la dernière correspond au produit du nombre d'enregistrements par le nombre de règles. La dernière colonne indique le nombre de règles ayant été appliquées à la variable en question.

Dans l'exemple utilisé dans la présente section, la variable x_1 était visée 11 fois par une règle pour laquelle le système a attribué le code d'état PASS: à savoir les règles (1), (4), (5) et (7) dans le cas de l'enregistrement 1; les règles (1), (4) et (5), dans le cas de l'enregistrement 2; enfin, les règles (1) et (4), dans le cas des enregistrements 3 et 4. Par ailleurs, la variable x_1 était visée deux fois par une règle pour laquelle le système a attribué le code MISS (soit la règle (7) dans le cas des enregistrements 2 et 4) et trois fois par une règle pour laquelle le système a attribué le code FAIL (les règles (5) et (7) dans le cas de l'enregistrement 3, et la règle (5) pour ce qui est de l'enregistrement 4). Étant donné qu'il y a trois règles qui ne visaient pas la variable x_1 , on obtient un total de 12 cas (3 règles x 4 enregistrements) où les règles de vérification ne s'appliquaient pas à cette variable. La dernière colonne indique le nombre de règles qui visaient la variable x_1 . Comme il y a quatre enregistrements dans l'exemple, il devrait y avoir 16 cas (4 x 4) où les règles de vérification s'appliquaient à la variable x_1 , et ce chiffre devrait toujours être égal à la somme des chiffres figurant dans les trois premières colonnes.

À partir du tableau 2-1, l'utilisateur peut juger si certaines variables ont tendance plus souvent que d'autres à être visées par des règles pour lesquelles le système attribue le code MISS ou FAIL.

TABLEAU 2-2

Totaux des observations avec l'état pass (succès), miss (manque) ou fail (échec) et pour lequel le champ j a contribué à l'état global de l'observation

FIELDID	OBS_PASSED	OBS_MISSED	OBS_FAILED	OBS_NOT_APPLICABLE
x_1	1	1	2	0
x_2	1	1	1	1
x_3	1	1	1	1

Le tableau 2-2 indique le nombre de fois où chaque variable a eu une incidence sur le code d'état général attribué à l'enregistrement. Aux fins de cette totalisation, le système attribue les codes d'état général PASS, MISS et FAIL à chaque variable suivant les règles suivantes.

- Si l'état général de l'enregistrement est PASS, l'enregistrement a satisfait à toutes les règles de vérification; les champs doivent donc tous contenir des valeurs valides et le système leur a tous attribué le code d'état PASS pour les fins de ce tableau.
- Si l'état général de l'enregistrement est MISS, les seuls codes d'état ayant pu être attribués pour les diverses règles de vérification sont les codes MISS et PASS. Le système attribue le code MISS aux variables visées par les règles pour lesquelles il a attribué le code d'état MISS et le code NOT APPLICABLE (SANS OBJET), aux variables qui ne sont visées par aucune de ces règles.
- Si l'état général de l'enregistrement est FAIL, le système doit avoir

attribué le code d'état FAIL pour au moins une règle et il peut avoir attribué le code d'état MISS pour une ou plusieurs règles. Le système attribue le code FAIL aux variables visées par les règles pour lesquelles il a attribué le code d'état FAIL et le code NOT_APPLICABLE, aux variables qui ne sont visées par aucune de ces règles.

Chaque enregistrement est compté une fois pour chaque variable, de sorte que le total pour la ligne est égal au nombre d'enregistrements. Ainsi, étant donné que l'enregistrement 1 a satisfait à toutes les règles de vérification, il est comptabilisé sous la rubrique « OBS_PASSED » vis-à-vis de toutes les variables. L'enregistrement 2 s'étant vu attribuer le code d'état général MISS et tous ses champs étant visés par au moins une règle pour laquelle le système a attribué l'état MISS, il est comptabilisé sous la rubrique « OBS_MISSED » vis-à-vis de toutes les variables, même si seulement x_3 était réellement manquante dans l'enregistrement de données. Pour ce qui est de l'enregistrement 3, son état général est FAIL et chacun de ses champs est visé par au moins une règle pour laquelle le système a attribué le code d'état FAIL. C'est pourquoi cet enregistrement est comptabilisé sous la rubrique « OBS_FAILED » vis-à-vis de toutes les variables. L'état général de l'enregistrement 4 est aussi FAIL. Sa variable x_1 est visée par la règle (5), pour laquelle le système a attribué le code FAIL; l'enregistrement 4 est donc comptabilisé dans la colonne sous la rubrique « OBS_FAILED », vis-à-vis de x_1 . Les deux autres champs de cet enregistrement n'étant visés par aucune règle pour laquelle le système a attribué le code d'état FAIL, cet enregistrement est comptabilisé dans la colonne sous la rubrique « OBS_NOT_APPLICABLE », vis-à-vis des variables x_2 et x_3 .

À partir du tableau 2-2, l'utilisateur peut déterminer si certaines variables tendent plus que d'autres à contribuer au fait que le système attribue un code d'état général MISS ou FAIL aux enregistrements. Il faut noter que dans ce tableau, toutes les variables visées par une règle pour laquelle le système a attribué le code d'état FAIL ou MISS sont comptabilisées comme contribuant au fait que l'état général des enregistrements soit FAIL ou MISS. La procédure de localisation des erreurs repérera probablement un sous-ensemble de ces champs devant faire l'objet d'une imputation et permettra aux autres champs visés par la règle de vérification de demeurer tels quels.

Utilisation des statistiques sommaires sur les règles de vérification

Cette procédure est la première dans lequel il y a une interaction entre les règles de vérification linéaires et les données fournies par les répondants. Dans la procédure d'analyse des règles de vérification, les règles linéaires sont examinées sans qu'on tienne compte des données. Dans la procédure de détection des valeurs aberrantes, qui peut être exécuté avant ou après la production des statistiques sommaires sur les règles de vérification, les données sont vérifiées suivant une règle statistique dont les paramètres sont déterminés à partir des données elles-mêmes, et non à partir des règles linéaires spécifiées par l'utilisateur.

Les statistiques sommaires sur les règles de vérification trouvent trois utilisations principales, se présentant chacune à une étape différente du processus de vérification et d'imputation. Ces utilisations et les étapes auxquelles elles se présentent sont examinées tour à tour.

Premièrement, ces statistiques permettent à l'utilisateur d'évaluer l'à-propos de règles individuelles ou d'un groupe de règles en observant les taux de rejet enregistrés lorsque les règles sont appliquées à un ensemble d'enregistrements de données. Un taux de rejet élevé pour une règle particulière peut indiquer à l'utilisateur qu'il doit modifier cette dernière en changeant les

constantes ou en modifiant la forme de la règle. Si ces modifications sont apportées, il faut alors changer la règle, régénérer la forme canonique et soumettre le nouveau groupe de règles à la procédure d'analyse des règles de vérification. Il est également possible que les anomalies observées dans les statistiques sommaires sur les règles de vérification ne soient pas attribuables à des problèmes relatifs au groupe de règles lui-même. Par exemple, si une variable est en cause dans un fort pourcentage de rejets à la vérification, il est possible qu'il y ait lieu de revoir les définitions utilisées dans le questionnaire ou les méthodes de collecte des données mises en œuvre sur le terrain. Bien sûr, il se pourrait que cette révision soit impossible à effectuer au cours des dernières étapes d'une enquête.

Lorsqu'on produit les tableaux de statistiques sommaires sur les règles de vérification en vue d'améliorer les règles, il est probable que les données d'enquête définitives ne soient pas encore disponibles. Il se peut que les enregistrements utilisés dans cette procédure soient des données réelles provenant d'une étude pilote ou de sources de données antérieures. Il est aussi possible d'avoir recours à des données qui ont été produites par l'utilisateur à l'extérieur de Banff.

Deuxièmement, les statistiques sommaires sur les règles de vérification sont utilisées pour consigner l'état des données fournies par les répondants à mesure que ces dernières sont entrées dans Banff et pour estimer le nombre d'enregistrements qui seront rejettés au cours de la localisation des erreurs. Ces renseignements servent à prévoir combien d'heures machine seront nécessaires pour l'exécution de la procédure de localisation des erreurs. Cette étape est exécutée après que les données fournies par les répondants ont été reçues et que les règles de vérification ont été arrêtées.

Enfin, ces statistiques servent à évaluer le processus d'imputation, soit après l'application de chacune des méthodes d'imputation, soit une fois l'imputation terminée. À cette fin, les tableaux produits à cette étape sont comparés à ceux qui ont été produits avant le début de l'imputation.

L'utilisateur est invité à consulter le *GEIS Applications User's Guide* pour obtenir de plus amples renseignements concernant les tableaux et leur interprétation.

Règles de vérification pour le traitement des valeurs négatives

La formulation des règles de vérification pour le traitement des valeurs négatives peut poser les défis qui peuvent produire les résultats imprévus si l'utilisateur ne les anticipe pas en avance. Cette situation demande une attention spéciale. Pour plus d'information et des exemples, voyez le document « Spécification des règles de vérification avec des données négatives dans Banff ». (Équipe de soutien de Banff, 2006).

4. PROC OUTLIER – DÉTECTION DES VALEURS ABERRANTES

Objet

Cette procédure utilise deux méthodes pour repérer les valeurs aberrantes, une décrite par Hidiroglou et Berthelot (1986) et l'autre, la méthode de l'écart-sigma développée à Statistique Canada dans les années 1990. Contrairement aux règles de vérification linéaires, elle permet de comparer les valeurs prises par certaines variables d'un enregistrement à l'autre plutôt que divers champs à l'intérieur de chaque enregistrement. L'utilisateur a aussi la possibilité de repérer les valeurs qui ne sont pas suffisamment extrêmes pour être considérées comme erronées, mais qui sont assez exceptionnelles pour ne pas être utilisées ultérieurement dans les procédures d'imputation. On peut limiter la détection des valeurs aberrantes à certaines variables; il n'est pas nécessaire de traiter toutes les variables.

Options offertes dans Proc Outlier

La procédure de détection des valeurs aberrantes peut repérer deux types de valeurs. Les premières, dites **valeurs aberrantes à imputer (ODI)**, sont des valeurs qui sont tellement différentes des autres valeurs prises par la même variable qu'elles peuvent être considérées comme erronées et qu'elles doivent être imputées dans une procédure ultérieure. Les secondes, dites **valeurs aberrantes à exclure (ODE)**, sont des valeurs qui ne sont pas suffisamment extrêmes pour être considérées comme erronées, mais qui sont assez exceptionnelles pour que l'utilisateur puisse songer à ne pas s'en servir ultérieurement dans la procédure d'imputation par enregistrement donneur ou à les exclure du calcul des paramètres utilisés par la procédure d'imputation par estimateur. Les paramètres sont définis de telle manière que la procédure de détection des valeurs aberrantes repérera les valeurs aberrantes à imputer ou les valeurs aberrantes à exclure, ou les deux. Dans les deux méthodes, Hidiroglou et Berthelot (1986) ainsi que l'écart-sigma, les limites utilisées pour définir ces valeurs ne sont pas fixes, mais elles sont fonction de paramètres précisés par l'utilisateur et des données elles-mêmes. Dans le fichier SAS de sortie qui contient les statuts des champs, les valeurs ODI sont écrites comme FTI (« field to impute », ou champ à imputer), et les valeurs ODE sont écrites comme FTE (« field to exclude », ou champ à exclure).

Les deux méthodes, Hidiroglou et Berthelot ainsi que l'écart-sigma, peuvent repérer les valeurs aberrantes parmi les valeurs à droite (ODER, ODIR), parmi les valeurs à gauche (ODEL, ODIL), ou des deux côtés. Le paramètre SIDE permet de spécifier le ou les côtés pour lesquels on souhaite repérer les valeurs aberrantes.

Le paramètre MINOBS permet de spécifier un nombre minimum d'enregistrements requis par groupe défini par l'énoncé BY pour effectuer la détection de valeurs aberrantes. MINOBS doit être supérieur ou égal à 3 pour la méthode Hidiroglou et Berthelot, 5 pour l'écart-sigma. Il est à noter que le système ne détectera aucune valeur aberrante si le nombre d'enregistrements est égal à 3 pour la méthode Hidiroglou et Berthelot. Il faut être plus prudent avec les résultats lorsqu'il y a moins de 10 enregistrements. Il est donc recommandé de mettre une valeur supérieure ou égale à 10 pour le paramètre MINOBS.

Le paramètre REJECTZERO va éliminer les valeurs nulles avant d'appliquer la détection de valeurs aberrantes. Le paramètre contraire, ACCEPTZERO, permettra aux valeurs nulles d'être utilisées dans les calculs impliquant les données courantes (voir le paragraphe ci-après). REJECTZERO est la valeur implicite pour la détection de valeurs aberrantes utilisant des ratios et

celle utilisant des tendances historiques. ACCEPTZERO est la valeur implicite pour la détection de valeurs aberrantes avec des données courantes. Les valeurs nulles et négatives ne sont jamais incluses dans la détection de valeurs aberrantes utilisant des ratios ni celle utilisant des tendances historiques.

La détection de valeurs aberrantes peut se faire de trois façons, autant avec la méthode Hidiroglou et Berthelot que l'écart-sigma : en utilisant des données courantes, des ratios ou des tendances historiques. Le choix approprié dépend de la nature des données à l'étude. La méthode Hidiroglou et Berthelot est décrite ci-après, suivie de la méthode de l'écart-sigma.

La méthode Hidiroglou et Berthelot

Si on ne dispose que de données relatives à une seule période et qu'il n'y a pas de bonne variable auxiliaire, il faut alors employer les données courantes, avec lesquelles le système compare chaque valeur de la variable choisie à des limites qui ont été calculées à partir des valeurs prises par cette variable dans les autres enregistrements. S'il y a une variable auxiliaire sûre, l'utilisateur peut analyser la variable choisie avec des ratios. De cette façon, on compare une fonction du ratio de la variable choisie et la variable auxiliaire à des limites fondées sur la même fonction pour les autres enregistrements. Enfin, en utilisant des tendances historiques, un cas particulier de la méthode Hidiroglou et Berthelot utilisée avec des ratios, la valeur auxiliaire est simplement la valeur historique de la variable choisie. Donc, l'utilisateur peut choisir cette façon de détecter des valeurs aberrantes plutôt qu'avec des données courantes ou des ratios s'il y a des données historiques.

Description de la méthode Hidiroglou et Berthelot utilisant des données courantes

Voici les étapes du traitement de chaque variable choisie lorsqu'on utilise des données courantes.

- Le système calcule le premier quartile, Q1, la médiane, M, et le troisième quartile, Q3, de la variable faisant l'objet du traitement. (Voir l'annexe A pour obtenir une définition de la médiane et des quartiles ainsi qu'une description de la méthode utilisée par Banff pour les calculer.) Si aucune exclusion n'a été précisée, ces valeurs sont calculées à partir de tous les enregistrements faisant partie du groupe de données. Sinon, les calculs sont fondés sur le sous-ensemble d'enregistrements qui reste après que tous les enregistrements qui satisfont aux critères d'exclusion ont été supprimés.
- Il calcule ensuite d_{Q1} et d_{Q3} , qui correspondent habituellement aux distances entre la médiane et les premier et troisième quartiles respectivement. Chacune de ces distances est remplacée par une valeur implicite précisée par l'utilisateur lorsqu'elle est inférieure à cette valeur implicite. Cela se produit lorsque la médiane est tellement rapprochée d'un quartile, ou des deux, que la ou les distances calculées seraient très peu élevées. La valeur implicite est une fonction de la médiane et du multiplicateur de distance minimale, paramètre spécifié par l'utilisateur, désigné par la lettre A dans les équations suivantes:.

$$d_{Q1} = \text{Max} (M - Q1, | A * M |)$$

$$d_{Q3} = \text{Max} (Q3 - M, | A * M |)$$

- Le système calcule ensuite les intervalles à l'intérieur desquels se situent les valeurs à imputer et les valeurs à exclure. Ces intervalles sont une fonction de d_{Q1} , d_{Q3} et des

multiplicateurs précisés par l'utilisateur pour l'intervalle des valeurs à imputer et pour l'intervalle des valeurs à exclure. Ces multiplicateurs sont désignés par C_I et C_E dans les équations suivantes qui définissent les régions ODI et ODE. L'utilisateur peut préciser les valeurs de C_I , de C_E , ou des deux multiplicateurs. Si les deux multiplicateurs sont spécifiés, C_I doit alors être supérieur à C_E . Si C_I n'est pas spécifié, aucune valeur ne sera désignée comme étant une valeur aberrante à imputer, bien qu'il puisse y avoir des valeurs aberrantes à exclure. Si C_E n'est pas spécifié, aucune valeur ne sera désignée comme étant une valeur aberrante à exclure, bien qu'il puisse y avoir des valeurs aberrantes à imputer.

$$\text{ODI : imputer si } \begin{cases} x_i < M - C_I d_{Q1} & \text{ou} \\ x_i > M + C_I d_{Q3} \end{cases}$$

$$\text{ODE : exclure si } \begin{cases} M - C_I d_{Q1} \leq x_i < M - C_E d_{Q1} & \text{ou} \\ M + C_I d_{Q3} \geq x_i > M + C_E d_{Q3} \end{cases}$$

- Enfin, le système repère les enregistrements qui se situent à l'intérieur des intervalles des valeurs FTI et FTE, puis il écrit ces statuts au fichier de sortie.

Exemple d'application de la méthode Hidiroglou et Berthelot utilisant des données courantes

Supposons qu'on ne dispose pour x , la variable devant être examinée par la procédure de détection des valeurs aberrantes, que de données courantes dont la distribution de fréquence correspond à celle qui est illustrée à la figure 4.1, et qu'aucun enregistrement ne doit être exclu des calculs. Les 24 valeurs que prend x sont, par ordre croissant :

-1, 4, 7, 7, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 10, 10, 11, 11, 11, 12, 13, 13, 15 et 19.

Comme il est décrit à l'annexe A, le premier quartile est égal à la somme du produit de la sixième valeur observée par .75 et du produit de la septième valeur observée par .25, la médiane est la moyenne des douzième et treizième valeurs, et le troisième quartile est égal à la somme du produit de la dix-huitième valeur observée par .25 et du produit de la dix-neuvième valeur observée par .75.

La médiane est 9, les premier et troisième quartiles sont 8 et 11 respectivement, et les distances entre la médiane et les quartiles sont 1 et 2. Le système calcule les limites en employant les valeurs spécifiées par l'utilisateur ($C_L = 6$ et $C_E = 4$) pour les multiplicateurs utilisés pour les valeurs à imputer et pour les valeurs à exclure respectivement. Ainsi, sont désignées comme étant des valeurs à imputer toutes les valeurs qui s'écartent de la médiane de plus de six fois la distance entre celle-ci et le quartile correspondant. Dans le présent exemple, l'observation ayant une valeur de 1 serait désignée comme étant une valeur aberrante à imputer (ODI) parce qu'elle s'écarte de la médiane d'une distance supérieure à six fois la distance entre cette dernière et le premier quartile. Aucune des valeurs ne serait désignée comme nécessitant une imputation parce qu'elle est trop élevée, car il n'existe aucune valeur qui s'écarte de la médiane de plus de six fois la distance entre cette dernière et le troisième quartile.

$$9 - (6 * 1) = 3 \\ 9 + (6 * 2) = 21$$

imputer si $\begin{cases} x_i < 3 \text{ ou} \\ x_i > 21 \end{cases}$

Les valeurs qui s'écartent de la médiane d'une distance inférieure à six fois la distance entre cette dernière et les quartiles mais supérieure à quatre fois la même distance sont désignées comme étant des valeurs à exclure. Dans cet exemple, les observations ayant des valeurs de 4 et de 19 seraient donc désignées comme étant des valeurs aberrantes à exclure.

$$9 - (4 * 1) = 5 \\ 9 + (4 * 2) = 17$$

exclure si $\begin{cases} 3 \leq x_i < 5 \text{ ou} \\ 21 \geq x_i > 17 \end{cases}$

Les limites qui sont utilisées lorsqu'on emploie la méthode Hidiroglou et Berthelot avec des données courantes sont linéaires par rapport à la variable faisant l'objet de l'examen; ainsi, au lieu d'exécuter la procédure de détection des valeurs aberrantes dans le mode de mise à jour, l'utilisateur pourrait créer des règles de vérification à l'aide des limites calculées par cette procédure, pour ensuite intégrer ces règles directement dans l'ensemble de règles de vérification linéaires.

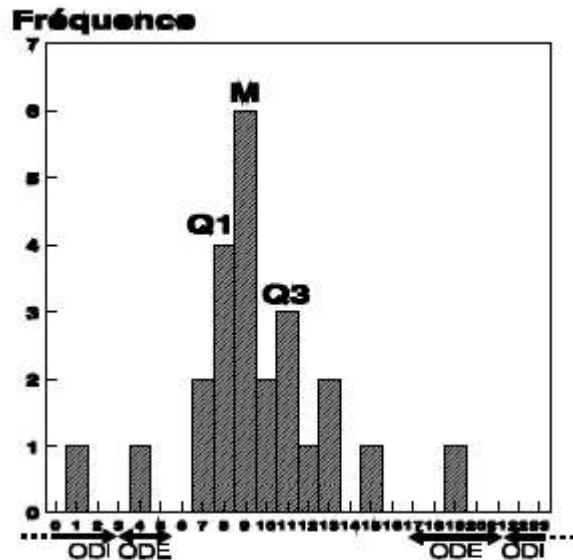


Figure 4.1 Exemple avec des données courantes

Description de la méthode Hidirogloou et Berthelot utilisant des ratios

Voici les étapes du traitement de chaque variable lorsqu'on utilise des ratios.

- Pour chaque enregistrement dont $x_i > 0$ et $y_i > 0$, le système calcule $r_i = \frac{x_i}{y_i}$, le rapport de x_i , la valeur courante de l'enregistrement i, à y_i , la valeur de la variable auxiliaire de l'enregistrement i.
- Il transforme ensuite les valeurs r de manière à ce qu'une diminution de n fois corresponde à une augmentation de n fois.

$$s_i = \begin{cases} 1 - \frac{r_M}{r_i} & 0 < r_i < r_M \\ \frac{r_i}{r_M} - 1 & r_i \geq r_M \end{cases} \quad \text{où } r_M \text{ est la médiane des quotients } r_i$$

- Il calcule enfin l'effet, e_i , de chaque enregistrement.

$$e_i = s_i [\max(x_i, y_i)]^{\exp}$$

Des calculs semblables à ceux qui sont exécutés avec des données courantes sont ensuite effectués sur ces valeurs e transformées. L'utilisateur peut donner à l'exposant (*exp*) toute valeur comprise entre 0 et 1. Lorsque l'exposant est égal à 0, toutes les variations relatives sont traitées de la même façon, quelle que soit la taille de l'unité, tandis que lorsque l'exposant est égal à 1, une plus grande importance est accordée aux petits écarts enregistrés pour les grandes unités.

- Le système calcule le premier quartile, la médiane et le troisième quartile des valeurs e transformées de la variable faisant l'objet du traitement. Si aucune exclusion n'a été précisée, il calcule ces valeurs à partir de tous les enregistrements dont les données courantes et historiques sont plus grandes que zéro. Sinon, les calculs sont fondés sur le sous-ensemble d'enregistrements qui reste après que tous les enregistrements qui satisfont aux critères d'exclusion spécifiés par l'utilisateur ont été supprimés.
- Il calcule ensuite d_{Q1} et d_{Q3} pour les valeurs e transformées. Ces distances correspondent habituellement aux distances entre la médiane et les premier et troisième quartiles respectivement. Chacune de ces distances est remplacée par une valeur implicite précisée par l'utilisateur lorsqu'elle est inférieure à cette valeur implicite. Cela se produit lorsque la médiane est tellement rapprochée d'un quartile des valeurs e transformées, ou des deux, que la ou les distances calculées seraient très peu élevées. La valeur implicite est une fonction de la médiane et du multiplicateur de distance minimale, paramètre spécifié par l'utilisateur, désigné par la lettre A dans les équations suivantes :

$$d_{Q1} = \text{Max}(M - Q1, |A * M|)$$

$$d_{Q3} = \text{Max}(Q3 - M, |A * M|)$$

- Le système calcule ensuite les intervalles à partir des valeurs transformées. Ces intervalles serviront à déterminer les valeurs à imputer et les valeurs à exclure. Ces intervalles sont une fonction des distances entre la médiane et les quartiles et des multiplicateurs précisés par l'utilisateur pour l'intervalle des valeurs à imputer et pour l'intervalle des valeurs à exclure. Ces multiplicateurs sont désignés par C_I et C_E dans les équations suivantes qui définissent les régions ODI et ODE. L'utilisateur peut préciser les valeurs de C_I , de C_E , ou des deux multiplicateurs. Si les deux multiplicateurs sont spécifiés, C_I doit être supérieur à C_E . Si C_I n'est pas spécifié, aucune valeur ne sera désignée comme étant une valeur aberrante à imputer, bien qu'il puisse y avoir des valeurs aberrantes à exclure. Si C_E n'est pas spécifié, aucune valeur ne sera désignée comme étant une valeur aberrante à exclure, bien qu'il puisse y avoir des valeurs aberrantes à imputer.

$$\text{ODI: impute si } \begin{cases} e_i < M - C_I d_{Q_1} \text{ ou} \\ e_i > M + C_I d_{Q_3} \end{cases}$$

$$\text{ODE: exclure si } \begin{cases} M - C_I d_{Q_1} \leq e_i < M - C_E d_{Q_1} \text{ ou} \\ M + C_I d_{Q_3} \geq e_i > M + C_E d_{Q_3} \end{cases}$$

- Enfin, le système repère les enregistrements qui se situent à l'intérieur des intervalles des valeurs FTI et FTE, puis il écrit ces statuts au fichier de sortie.

Pour la détection des valeurs aberrantes en utilisant des ratios, les limites calculées sont linéaires par rapport à la variable transformée, mais non linéaires par rapport à la variable initiale, à moins que l'exposant 0 ait été utilisé. Dans l'exemple utilisant des tendances historiques ci-après, on peut constater qu'il s'agit de la méthode Hidiroglou et Berthelot utilisant des ratios dans le cas particulier où les valeurs de la variable auxiliaire sont les valeurs historiques de la variable choisie.

Description de la méthode Hidiroglou et Berthelot utilisant des tendances historiques

Il s'agit d'un cas particulier de la méthode Hidiroglou et Berthelot utilisant des ratios où la variable auxiliaire y_i est la variable x_i choisie d'une période antérieure, c'est-à-dire $x_{i(t-1)}$. La période courante est représentée par l'indice t et la période précédente par l'indice $(t-1)$. Les étapes à suivre avec des tendances historiques sont les mêmes que celles avec des ratios avec la variable choisie représentée par x_{it} et sa valeur historique par $x_{i(t-1)}$, où t et $t-1$ représente les périodes de

temps. Donc, le ratio pour l'enregistrement i est $r_i = \frac{x_{it}}{x_{i(t-1)}}$.

Exemple d'application de la méthode Hidiroglou et Berthelot utilisant des tendances historiques

Prenons par exemple 22 observations effectuées à l'occasion de chacune de deux périodes de référence. Au lieu de reproduire tous les calculs effectués par Banff, nous avons présenté sous forme de graphiques les résultats de la détection des valeurs aberrantes exécutée suivant diverses combinaisons de paramètres. Le tableau qui suit indique les valeurs des données fournies par les répondants (x_t et $x_{(t-1)}$), ainsi que les valeurs r , rapport de la période courante à la période précédente, et les valeurs transformées s . Les résultats de la détection des valeurs aberrantes sont présentés aux figures 4.3a, 4.3b, 4.4a, 4.4b, 4.5a et 4.5b. Enfin, la figure 4.6 indique les résultats

de l'application de la méthode Hidiroglou et Berthelot utilisant des données courantes exclusivement aux données obtenues pour la période courante.

Les observations suivantes sont classées par ordre croissant de la valeur r :

Ident.	x_t	$x_{(t-1)}$	r	s	Ident.	x_t	$x_{(t-1)}$	r	s
01	60	160	0.375	-1.667	13	180	180	1.000	0.000
02	15	35	0.429	-1.333	14	200	200	1.000	0.000
03	150	192	0.781	-0.280	15	90	85	1.059	0.059
04	130	150	0.867	-0.154	16	100	85	1.176	0.176
05	40	45	0.889	-0.125	17	165	140	1.179	0.179
06	160	175	0.914	-0.094	18	30	21	1.429	0.429
07	70	75	0.933	-0.071	19	300	200	1.500	0.500
08	70	71	0.986	-0.014	20	100	50	2.000	1.000
09	40	40	1.000	0.000	21	195	97	2.010	1.010
10	62	62	1.000	0.000	22	160	60	2.667	1.667
11	75	75	1.000	0.000	23	5	0		exclu
12	100	100	1.000	0.000	24	-10	5		exclu

On peut voir l'effet de la transformation s en examinant les enregistrements 01 et 22. Dans l'enregistrement 01, la valeur de x a diminué pour descendre de 160 pour la période (t-1) à 60 pour la période t; dans l'enregistrement 22, la valeur de x a augmenté pour passer de 60 à 160 au cours de la même période. Bien que ces variations soient similaires, les valeurs r non transformées correspondantes, 0.375 et 2.667, ne sont pas à la même distance de la valeur r de la médiane, qui est égale à 1.000. Par suite de leur transformation en valeurs s de -1.667 et de 1.667, ces variations similaires correspondent à un même écart de part et d'autre de 0.0, la médiane des valeurs s.

L'enregistrement 23 est exclu de la méthode Hidiroglou et Berthelot utilisant des tendances historiques en raison de sa valeur zéro dans le fichier de données historiques. L'enregistrement 24 est exclu en raison de sa valeur négative dans le fichier de données courantes.

Exemple d'application avec des tendances historiques - Exposant = 0

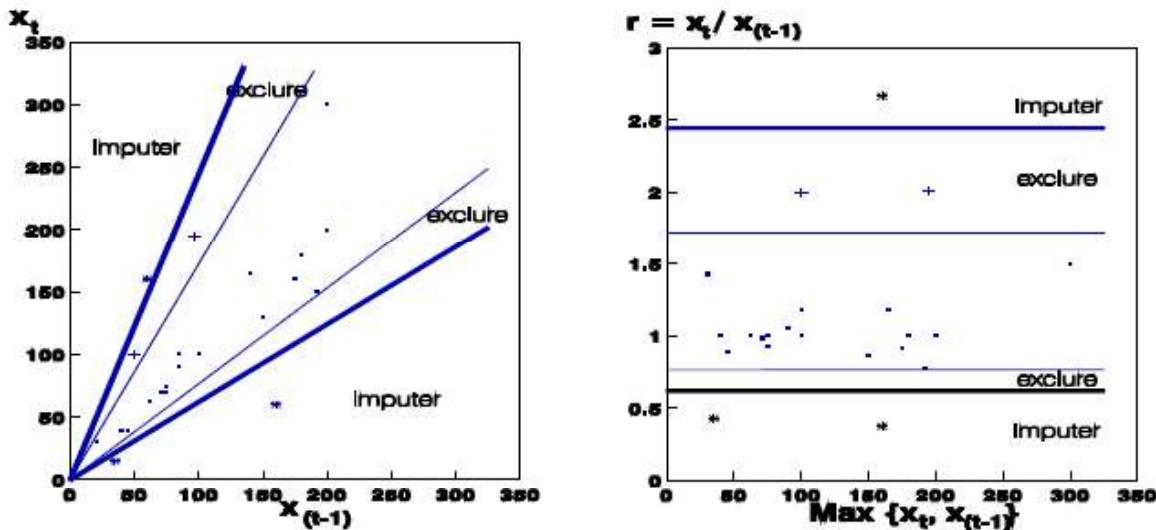


Figure 4.3a Observations courantes par rapport aux observations précédentes
Exposant = 0

Figure 4.3b Rapport des observations en fonction des valeurs maximales
Exposant = 0

La figure 4.3a représente graphiquement chaque observation courante, x_{It} , en fonction de la valeur historique correspondante, $x_{I(t-1)}$, tandis que la figure 4.3b représente graphiquement le rapport de ces deux valeurs en fonction du maximum qu'elles peuvent atteindre. Dans les deux figures, on a employé la méthode Hidiroglou et Berthelot utilisant des tendances historiques avec un exposant égal à 0 et des multiplicateurs de 6 et de 3 pour les limites d'imputation et d'exclusion. Les valeurs observées se trouvant à l'extérieur des traits forts doivent être imputées. Dans cet exemple, il y a trois de ces valeurs, chacune désignée par un « * ». Les valeurs situées entre le trait fort et le trait plus fin sont les valeurs à exclure. Dans cet exemple, il y a deux de ces valeurs, désignées par un « + ». On notera que les limites sont des droites dans les deux graphiques où on utilise un exposant égal à 0.

Exemple d'application avec des tendances historiques - Exposant = 1

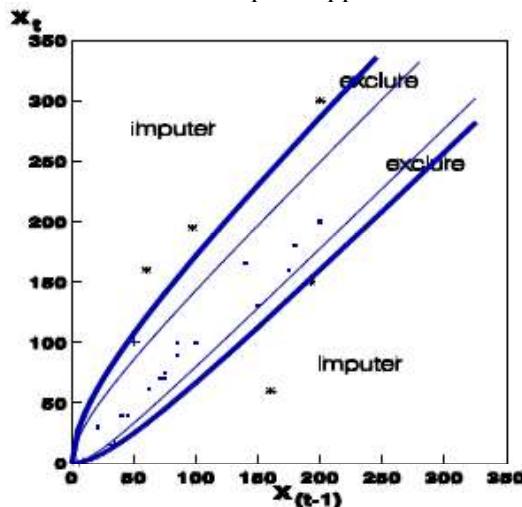


Figure 4.4a Observations courantes par rapport aux observations précédentes
Exposant = 1

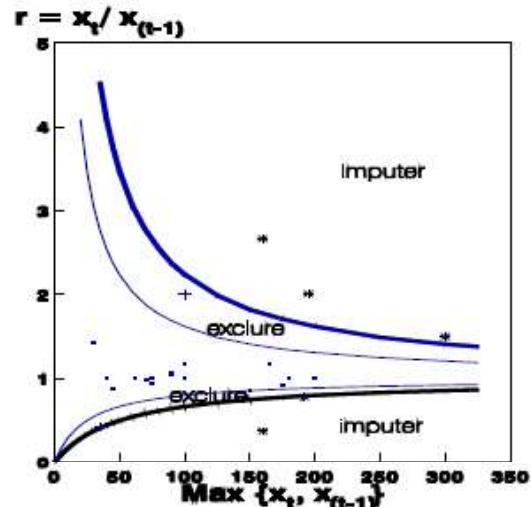


Figure 4.4b Rapport des observations en fonction des valeurs maximales
Exposant = 1

La figure 4.4a représente graphiquement chaque observation courante, x_{It} , en fonction de la valeur historique correspondante, $x_{I(t-1)}$, tandis que la figure 4.4b représente graphiquement le rapport de ces deux valeurs en fonction du maximum qu'elles peuvent atteindre. Dans les deux figures, on a employé la méthode des tendances historiques avec un exposant égal à 1 et des multiplicateurs de 6 et de 3 pour les limites d'imputation et d'exclusion. Les valeurs observées se trouvant à l'extérieur des traits forts doivent être imputées. Dans cet exemple, il y a cinq de ces valeurs, chacune désignée par un « * ». Les valeurs situées entre le trait fort et le trait plus fin sont les valeurs à exclure. Il y a deux de ces valeurs, désignées par un « + ». On notera que les limites ne sont pas des droites lorsqu'on utilise un exposant de 1, mais plutôt des courbes qui permettent les variations relatives plus élevées pour les petites unités.

Exemple d'application avec des tendances historiques - Comparaison des exposants

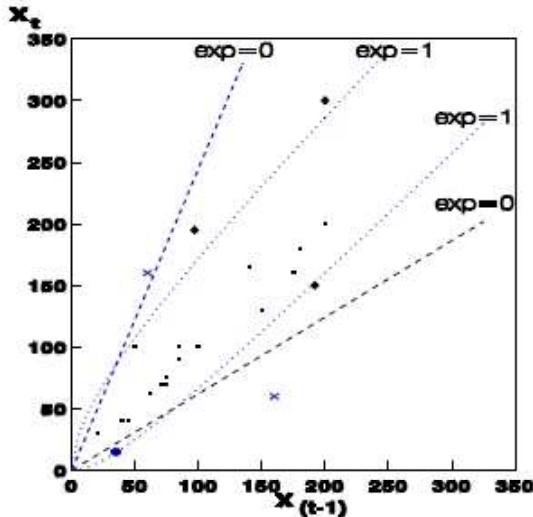


Figure 4.5a Observations courantes par rapport aux observations précédentes
Comparaison entre $\text{exp}=0$ et $\text{exp}=1$

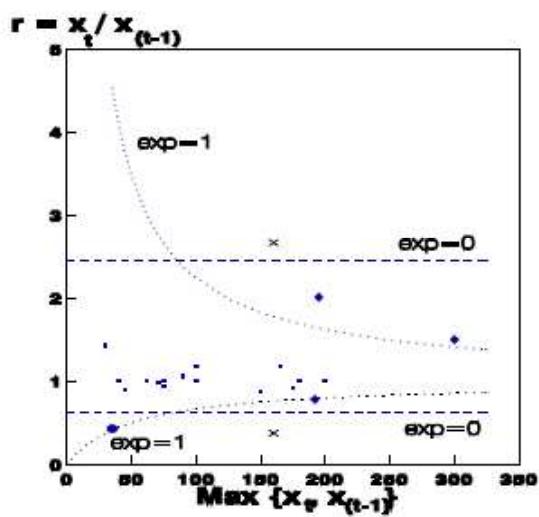


Figure 4.5b Rapport des observations en fonction des valeurs maximales
Comparaison entre $\text{exp}=0$ et $\text{exp}=1$

Pour faciliter la comparaison des résultats de l'utilisation des deux exposants différents, les traits forts indiquant quelles valeurs observées doivent être imputées dans les figures 4.3a et 4.4a sont réunis dans la figure 4.5a, tandis que les traits forts des figures 4.3b et 4.4b sont réunis dans la figure 4.5b. Les traits discontinus représentent les limites qui seraient appliquées lorsque l'exposant 0 est utilisé et les traits pointillés représentent les limites qui seraient appliquées lorsque l'exposant est 1. Les limites d'exclusion ne sont pas indiquées. Les deux observations indiquées par un « x » ont été désignées comme devant être imputées, quel que soit l'exposant utilisé. Les trois observations indiquées par un « ♦ » ont été désignées comme devant être imputées lorsque l'exposant 1 est utilisé, mais non lorsque l'exposant est 0. L'observation indiquée par un « ● » a été désignée comme devant être imputée lorsque l'exposant 0 est employé, mais non lorsque l'exposant est 1.

Il y a lieu de noter que lorsqu'on utilise l'exposant 1, le système tend à désigner un plus grand nombre de grandes unités comme étant des valeurs aberrantes à imputer, et à omettre quelques petites unités qui seraient des valeurs aberrantes à imputer si on utilisait l'exposant 0. Lorsque l'exposant est 1, les variations modestes d'une grande unité deviennent plus susceptibles d'être désignées comme nécessitant une imputation et les écarts moyens d'une petite unité deviennent plus susceptibles d'être acceptables. Cette situation s'explique du fait que la grandeur de l'observation elle-même a une incidence sur la valeur de r lorsque l'exposant est supérieur à 0, ce qui n'est pas le cas lorsque celui-ci est égal à 0. Comme on peut le voir dans les graphiques, lorsque l'exposant est égal à 1, les limites courbent vers l'extérieur pour inclure les variations relatives moyennes des petites unités, puis se resserrent pour n'admettre que des variations modestes des grandes unités. En utilisant un exposant dont la valeur se situe entre 0 et 1, on peut obtenir des limites dont la courbe est moins prononcée que celle des limites obtenues à l'aide de l'exposant 1.

Comparaison entre la méthode Hidiroglou et Berthelot utilisant des données courantes et utilisant des tendances historiques

Examinons maintenant les résultats de l'application de la méthode Hidiroglou et Berthelot utilisant des données courantes exclusivement aux valeurs x_t des données utilisées dans l'exemple de l'application avec des tendances historiques. Afin de traiter exactement le même groupe d'enregistrements, les enregistrements 23 et 24 doivent être exclus.

Les données x_t sont présentées ci-dessous classées par ordre croissant de la valeur x_t . À la figure 4.6, les mêmes données x_t sont représentées graphiquement à l'aide d'un histogramme.

Ident.	x_t	Ident.	x_t	Ident.	x_t	Ident.	x_t
02	15	07	70	20	100	17	165
18	30	08	70	04	130	13	180
05	40	11	75	03	150	21	195
09	40	15	90	22	160	14	200
01	60	12	100	06	160	19	300
10	62	16	100				

Paramètres spécifiés par l'utilisateur : $A = .05$ $C_I = 6$ $C_E = 3$

Données calculées par le système: $Q1 = 61.5$ $M = 100$ $Q3 = 161.25$

$$d_{Q1} = 38.5$$

$$d_{Q3} = 61.25$$

imputer si : $x_{it} < -131$ ou $467.50 < x_{it}$
 exclure si : $-131 \leq x_{it} < -15.5$ ou $283.75 < x_{it} \leq 467.50$

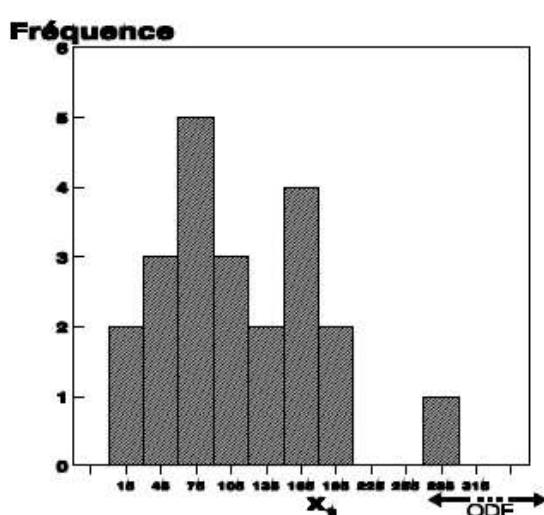


Figure 4.6 Histogramme des données courantes, x_t

Notons que, dans cet exemple, la méthode Hidiroglou et Berthelot utilisant des données courantes ne permettrait pas de désigner une valeur négative comme devant être imputée, à moins que cette valeur soit inférieure à -131. Les valeurs négatives qui ne s'écartent pas suffisamment de la médiane ne sont pas automatiquement désignées pour fin d'imputation dans cette procédure. Toutefois, si l'utilisateur continue le traitement avec Banff et exécute la localisation des erreurs avec l'option pour rejeter les valeurs négatives (section 5), alors toutes les valeurs négatives seront désignées comme devant être imputées parce que cette procédure ajoutera automatiquement pour chaque variable une règle de vérification fondée sur la positivité des valeurs.

Lorsque la méthode Hidiroglou et Berthelot utilisant des données courantes a été utilisée, la valeur 300 de l'enregistrement 19 a été désignée comme étant une valeur à exclure, et aucune valeur à imputer n'a été repérée. Ce résultat est différent de celui obtenu en employant la méthode Hidiroglou et Berthelot avec des tendances historiques, selon laquelle l'enregistrement 19 a été désigné comme nécessitant une imputation lorsque l'exposant était 1, mais n'a pas été désigné comme étant un champ à imputer ou à exclure lorsque l'exposant était 0.

La méthode de l'écart-sigma

La méthode de l'écart-sigma peut être utilisée avec des données courantes, des ratios ou des tendances historiques. Veuillez vous référer au premier paragraphe de la section portant sur la méthode Hidiroglou et Berthelot pour découvrir comment déterminer la façon appropriée d'utiliser la méthode de l'écart-sigma selon vos données.

Description de la méthode de l'écart-sigma utilisant des données courantes

Les étapes suivantes sont suivies pour chacune des variables sélectionnées.

- Le système calcule l'écart σ_x . La méthode de l'écart-sigma offre deux façons de calculer l'écart grâce à l'option SIGMA. La première façon est l'écart-type basé sur la moyenne (STD):

$$\sigma_{STD} = \sqrt{\frac{\sum (w_i x_i - \bar{w}_i \bar{x}_i)^2}{n - 1}}$$

où x est la valeur originale non pondérée de la variable X et w est le poids correspondant fourni grâce à l'option WEIGHT. La valeur implicite du poids est 1 si l'option WEIGHT n'est pas définie.

L'autre écart disponible est dérivé de l'écart médian absolu (MAD) qui se fonde sur la distance de chaque enregistrement à la médiane :

$$\sigma_{MAD} = 1,4826 * med(|w_i x_i - med_j(w_j x_j)|)$$

où la médiane intérieure $med_j(w_j x_j)$ est la médiane des n valeurs pondérées et, med est la médiane extérieure des écarts des n valeurs pondérées à la médiane $med_j(w_j x_j)$ en valeur absolue. Le facteur d'ajustement 1.4826 fait de σ_{MAD} un estimateur convergent de σ_{STD} pour une population normale (voir Rousseeuw et Leroy, page 202). Exemple : la médiane intérieure des valeurs (5, 10, 15, 20, 25) est 15. Les distances à la médiane en valeur absolue est, dans le même ordre, $|5-15|=10$, $|10-15|=5$, $|15-15|=0$, $|20-15|=5$, et $|25-15|=10$ qui, une fois ordonnées deviennent (0, 5, 5, 10, 10). L'écart médian absolu est la médiane de ces valeurs (médiane extérieure), ce qui donne 5. Par conséquent, $\sigma_{MAD} = 1,4826 * 5 = 7,413$ dans cet exemple.

L'écart σ_{STD} peut être largement influencé par les valeurs extrêmes contrairement à σ_{MAD} . Cela implique que les valeurs aberrantes potentielles influencent l'écart qui sert à les repérer. Il est recommandé d'utiliser σ_{MAD} pour cette raison. Voici un exemple qui

démontre l'effet des valeurs aberrantes sur les écarts. Les valeurs de la variable X suivent une distribution normale :

-27, -22, -21, -19, -16, -16, -15, -15, -12, -12, -8, -6, -5, -2, -2, -2, 1, 7, 8, 8, 9, 10, 14, 19, 24, 26, 29, 32, 36 et 45

Les écarts sont similaires pour ces valeurs : $\sigma_{STD} = 19,12$ et $\sigma_{MAD} = 19,27$. En modifiant les deux dernières valeurs de l'ensemble de données précédentes pour inclure deux valeurs extrêmes, soit

-27, -22, -21, -19, -16, -16, -15, -15, -12, -12, -8, -6, -5, -2, -2, -2, 1, 7, 8, 8, 9, 10, 14, 19, 24, 26, 29, 32, 136 et 145,

les écarts deviennent : $\sigma_{STD} = 39,20$ et $\sigma_{MAD} = 19,27$. L'écart σ_{STD} a doublé de valeur alors que σ_{MAD} est resté le même. Par conséquent, l'écart σ_{MAD} tend moins à être influencé par les valeurs extrêmes.

- Ensuite, le système calcule deux écarts-sigma, un pour repérer les valeurs ODE, un pour les ODI. Pour ce faire, l'écart calculé précédemment est multiplié par la valeur de l'option BETA_E (β_E) pour les ODE et par BETA_I (β_I) pour les ODI. Si l'option OUTLIERSTAT est utilisée, alors ces écarts-sigma vont apparaître dans le fichier de sortie sous les noms respectifs EXCL_SIGMAGAP et IMP_SIGMAGAP pour chaque groupe BY :

$$EXCL_SIGMAGAP = \beta_E \sigma_{STD} \text{ ou } \beta_E \sigma_{MAD}$$

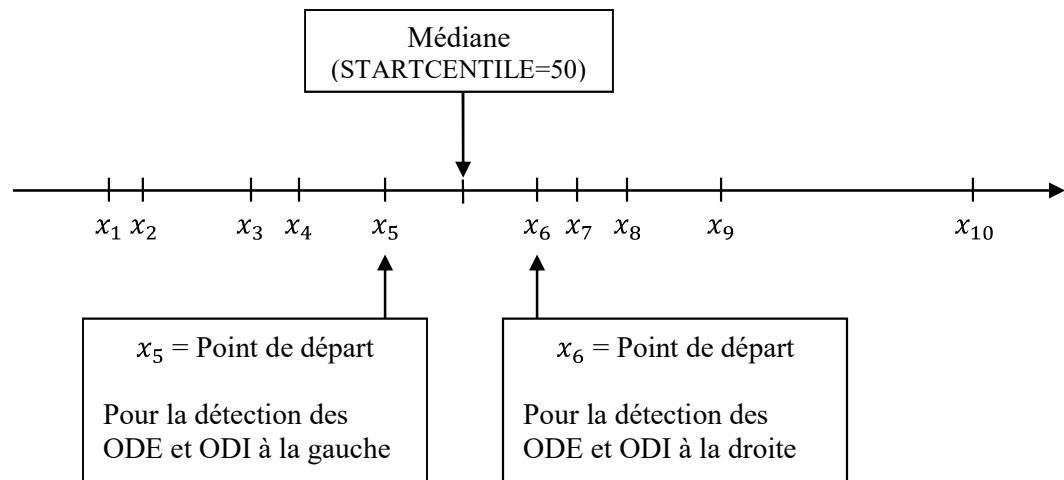
$$IMP_SIGMAGAP = \beta_I \sigma_{STD} \text{ ou } \beta_I \sigma_{MAD}$$

- Puis, le point de départ est trouvé. Cette fonction est unique à la méthode de l'écart-sigma. Bien que toutes les valeurs aient été utilisées jusqu'à présent pour calculer l'écart-sigma pour les valeurs aberrantes à exclure et l'écart-sigma pour les valeurs aberrantes à imputer, il est possible de limiter la détection de valeurs aberrantes à seulement une partie des enregistrements grâce à l'option STARTCENTILE. Les exemples ci-après permettent d'illustrer ce principe.

Soit les valeurs de la variable X pour 10 enregistrements déjà triés en ordre croissant : x_1, x_2, \dots, x_{10} . La valeur implicite de STARTCENTILE est 0 lorsque SIDE=LEFT (détection à gauche) ou SIDE=RIGHT (détection à droite). Supposons que la détection est effectuée vers la droite uniquement. Donc, sauf pour la plus petite valeur (x_1), toutes les autres valeurs peuvent devenir soit ODE ou ODI. Cependant, en définissant la valeur de l'option STARTCENTILE à 75, la détection des valeurs aberrantes sera appliquée à partir de la première valeur rencontrée au 75^{ème} centile inclusivement et en regardant vers la droite, x_8 dans cet exemple. Ainsi, la valeur de x_8 est le point de départ, ce qui veut aussi dire que seules les valeurs pour les enregistrements x_9 et x_{10} peuvent devenir ODE ou ODI. Il est à noter que si plusieurs enregistrements ont la même valeur que celle de l'enregistrement à la position du centile, Banff choisira comme point de départ l'enregistrement le plus à droite (voir l'exemple à la page 4-14). Si la détection est effectuée vers la gauche seulement, alors il faut trouver l'enregistrement au 75^{ème} centile

en partant de l'enregistrement ayant la plus grande valeur et en se dirigeant vers la gauche. Ici, x_3 est le point de départ pour la détection vers la gauche avec STARTCENTILE=75 et seules les valeurs x_2 et x_1 peuvent devenir ODE ou ODI. Encore une fois, si plusieurs enregistrements ont la même valeur que celle de l'enregistrement à la position du centile, Banff choisira comme point de départ l'enregistrement le plus à gauche parmi ceux-ci.

Examinons de nouveau les dix enregistrements précédents, soit un nombre pair d'enregistrements, mais avec l'option SIDE=BOTH. Dans ce cas, la détection est effectuée vers la gauche et vers la droite des données. La valeur minimum pour l'option STARTCENTILE lorsque la détection est effectuée des deux côtés est 50, soit la médiane. Voici les points de départs lorsque toutes les valeurs x_i sont différentes :



Donc, avec SIDE=BOTH et STARTCENTILE=50, le premier enregistrement en se dirigeant vers la droite à partir de la médiane est x_6 . C'est le point de départ pour la détection de valeurs aberrantes vers la droite. Le premier enregistrement en se dirigeant vers la gauche à partir de la médiane est x_5 qui est le point de départ pour la détection de valeurs aberrantes vers la gauche. La médiane est l'enregistrement du milieu lorsqu'il y a un nombre impair d'enregistrements. Cet enregistrement du milieu devient donc le point de départ autant pour la détection de valeurs aberrantes vers la droite que vers la gauche.

Il est important de noter que le point de départ ne peut jamais être ODE ni ODI. La valeur implicite de l'option STARTCENTILE lorsque SIDE=BOTH est 75.

- Enfin, le système calcule les écarts entre chaque valeur triée pour repérer les enregistrements qui tombent dans les intervalles ODI et ODE.

Pour la détection des valeurs aberrantes vers la droite, x_{i+1} ainsi que toutes les valeurs plus grandes ou égales à x_{i+1} sont ODE si $x_{i+1} - x_i > EXCL_SIGMAGAP$, où la valeur de i correspond à la position du point de départ et $i+1$ représente la position de l'enregistrement suivant vers la droite. Pour la détection des valeurs aberrantes vers la droite, x_{i+1} ainsi que toutes les valeurs plus grandes ou égales à x_{i+1} sont ODI dès que $x_{i+1} - x_i > IMP_SIGMAGAP$.

Pour la détection des valeurs aberrantes vers la gauche, x_{i-1} ainsi que toutes les valeurs plus petites ou égales à x_{i-1} sont ODE si $x_i - x_{i-1} > EXCL_SIGMAGAP$, où la valeur de i correspond à la position du point de départ et $i-1$ représente la position de l'enregistrement suivant vers la gauche. Pour la détection des valeurs aberrantes vers la gauche, x_{i-1} ainsi que toutes les valeurs plus petites ou égales à x_{i-1} sont ODI dès que $x_i - x_{i-1} > IMP_SIGMAGAP$.

Si un écart définit le même intervalle pour les ODE et les ODI alors il n'y aura que des champs à imputer (STATUS=FTI). Le système repère donc les enregistrements qui se situent à l'intérieur des intervalles ODE et ODI, puis inscrit les statuts FTE et FTI correspondants au fichier de sortie.

Premier exemple d'application de la méthode de l'écart-sigma utilisant des données courantes

Les données utilisées pour l'exemple de la méthode Hidirogloou et Berthelot avec des données courantes sont utilisées de nouveau ici à titre comparatif. Les 24 valeurs de x en ordre croissant sont :

-1, 4, 7, 7, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 10, 10, 11, 11, 11, 11, 12, 13, 13, 15 et 19.

L'option SIDE est définie avec la valeur BOTH pour effectuer la détection des valeurs aberrantes vers la gauche et vers la droite. L'option STARTCENTILE est définie à 75. Le 75^{ème} centile en partant de l'enregistrement ayant la plus petite valeur et en allant vers la droite est x_{18} qui a une valeur de 11. Puisqu'il y a trois enregistrements avec cette valeur, Banff se rend au dernier enregistrement de ce groupe vers la droite, soit x_{19} qui devient le point de départ ($x_i = x_{19}$) pour la détection vers la droite. Le premier écart entre deux valeurs qui sera calculé pour repérer les valeurs aberrantes vers la droite ($x_{i+1} - x_i$) est $x_{20} - x_{19}$, ce qui donne 12-11=1 qui sera comparé aux écarts-sigma d'exclusion et d'imputation. Le deuxième écart sera $x_{21} - x_{20}$ et ainsi de suite.

Le même procédé est appliqué vers la gauche. Le 75^{ème} centile en partant de l'enregistrement avec la plus grande valeur et en se dirigeant vers la gauche est x_7 , qui a une valeur de 8. Puisqu'il y a quatre enregistrements avec cette valeur, Banff se rend au dernier enregistrement de ce groupe vers la gauche, soit x_5 qui devient ainsi le point de départ ($x_i = x_5$) pour la détection vers la gauche. Le premier écart entre deux valeurs successives qui sera calculé pour repérer les valeurs aberrantes vers la gauche ($x_i - x_{i-1}$) est $x_5 - x_4$, ce qui donne 8-7=1 qui sera comparé aux écarts-sigma d'exclusion et d'imputation. Le deuxième écart sera $x_4 - x_3$ et ainsi de suite.

L'écart médian absolu qui sera utilisé pour calculer l'écart est 2,22 selon ces valeurs de x . Le multiplicateur pour l'exclusion qui sera utilisé est $\beta_E=1,5$ et le multiplicateur pour l'imputation est $\beta_I=3,0$.

L'écart-sigma pour l'exclusion est calculé et il sera utilisé pour détecter les ODE vers la gauche et vers la droite :

$$EXCL_SIGMAGAP = \beta_E \sigma_{MAD} = (1,5) * (2,22) = 3,33$$

Dès qu'un écart calculé entre deux valeurs successives est plus grand que 3,33, la plus grande de ces deux valeurs (pour la détection vers la droite) ou la plus petite de ces deux valeurs (pour la détection vers la gauche) est identifiée comme valeur à exclure (ODE). L'écart-sigma pour l'imputation est calculé et il sera utilisé pour détecter les ODI vers la gauche et vers la droite :

$$IMP_SIGMAGAP = \beta_I \sigma_{MAD} = (3,0) * (2,22) = 6,66$$

Dès qu'un écart calculé entre deux valeurs successives est plus grand que 6,66, la plus grande de ces deux valeurs (pour la détection vers la droite) ou la plus petite de ces deux valeurs (pour la détection vers la gauche) est identifiée comme valeur à imputer (ODI).

En d'autres termes, tout écart tombant dans l'intervalle (3,33 , 6,66] identifiera les ODE et tout écart plus grand que 6,66 identifiera les ODI. Par exemple, l'écart entre $x_{22}=13$ et $x_{23}=15$ est 2, qui est plus petit que 3,33 alors la valeur x_{23} n'est pas un ODE ni un ODI. Cependant, l'écart suivant entre $x_{23}=15$ et $x_{24}=19$ est 4 qui tombe dans l'intervalle (3,33 , 6,66] et donc, fait de la valeur x_{24} un ODE.

Dans cet exemple avec la méthode de l'écart-sigma, $x_1=-1$ est repérée comme valeur aberrante à exclure (ODE) vers la gauche et $x_{24}=19$ est un ODE vers la droite. Dans l'exemple précédent avec la méthode Hidiroglou et Berthelot utilisant les mêmes données, vers la gauche, $x_1=-1$ était une valeur aberrante à imputer (ODI) et $x_2=4$ était un ODE et, vers la droite, $x_{24}=19$ était un ODE.

Deuxième exemple d'application de la méthode de l'écart-sigma utilisant des données courantes
Supposons les 20 valeurs suivantes en ordre croissant pour la variable x :

5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 24, 24, 25, 25, 25, 25, 27, 28 et 100

Ici, nous souhaitons détecter les valeurs aberrantes en vérifiant les écarts entre toutes les valeurs et uniquement vers la droite. Donc, il faut utiliser l'option SIDE=RIGHT et l'option implicite STARTCENTILE=0. Alors, l'enregistrement avec la plus petite valeur parmi les données triées en ordre croissant, $x_1=5$, est le point de départ ($x_i = x_1$) et toutes les autres valeurs suivantes seront testées pour voir si elles deviennent ODE ou ODI. Les écarts-sigma seront basés de nouveau sur l'écart médian absolu (SIGMA=MAD). L'écart basé sur l'écart médian absolu calculé est 2,97. Tout comme pour l'exemple précédent, $\beta_E=1,5$ et $\beta_I=3,0$ sont définis.

$$EXCL_SIGMAGAP = \beta_E \sigma_{MAD} = (1,5) * (2,97) = 4,46$$

$$IMP_SIGMAGAP = \beta_I \sigma_{MAD} = (3,0) * (2,97) = 8,91$$

Dès qu'un écart calculé entre deux valeurs successives tombe dans l'intervalle (4,46 , 8,91], la plus grande de ces deux valeurs est identifiée comme valeur à exclure (ODE). Dès qu'un écart calculé

entre deux valeurs successives est plus grand que 8,91, la plus grande de ces deux valeurs est identifiée comme valeur à imputer (ODI).

Avec les données présentes, le premier écart supérieur à 4,46 et même à 8,91 est 17, soit entre les valeurs $x_{11}=7$ et $x_{12}=24$. Les deux valeurs 24 ainsi que toutes les valeurs supérieures à 24 sont par conséquent identifiées comme valeurs à imputer (ODI) puisque l'écart 17 est supérieur à l'écart-sigma pour l'imputation. Près de la moitié des valeurs ont été identifiées valeurs aberrantes à imputer parce qu'elles sont trop grandes. Ceci est dû au fait que les valeurs de l'ensemble de données forment deux classes distinctes de valeurs similaires, soit celles de 5 à 7 et celles supérieures à 23. En définissant STARTCENTILE=80 au lieu de la valeur implicite 0, seules les plus grandes valeurs seront testées. Dans ce cas, la seule valeur aberrante à imputer devient la valeur $x_{20}=100$.

Description de la méthode de l'écart-sigma utilisant des ratios

Les étapes suivantes sont suivies pour chacune des variables sélectionnées.

- Pour chaque enregistrement dont $x_i > 0$ et $y_i > 0$, le système calcule $r_i = \frac{x_i}{y_i}$, le rapport

de x_i , la valeur courante de l'enregistrement i, à y_i , la valeur de la variable auxiliaire de l'enregistrement i. Toutes les autres étapes sont les mêmes que celles appliquées en utilisant des données courantes. Il suffit de remplacer x_i par r_i dans les formules. Les étapes sont résumées ci-après et, pour plus de détails, veuillez consulter la section précédente portant sur la méthode de l'écart-sigma utilisant des données courantes.

- Le système calcule l'écart σ_r :

$$\sigma_{STD} = \sqrt{\frac{\sum \left(w_i r_i - \frac{\sum w_i r_i}{n} \right)^2}{n - 1}}$$

ou

$$\sigma_{MAD} = 1.4826 * med(|w_i r_i - med_j(w_j r_j)|)$$

- Ensuite, le système calcule deux écarts-sigma, un pour repérer les valeurs ODE, un pour les ODI pour chaque groupe de l'énoncé BY:

$$EXCL_SIGMAGAP = \beta_E \sigma_{STD} \text{ ou } \beta_E \sigma_{MAD}$$

$$IMP_SIGMAGAP = \beta_I \sigma_{STD} \text{ ou } \beta_I \sigma_{MAD}$$

- Puis, le système détermine le point de départ, d'après l'option STARTCENTILE, parmi les ratios r_i triés en ordre croissant : r_1, r_2, \dots, r_{10} .
- Ensuite, le système calcule les écarts entre chaque valeur triée (r_i) pour repérer les ratios qui tombent dans les intervalles ODI et ODE.

Pour la détection des valeurs aberrantes vers la droite, r_{i+1} ainsi que toutes les valeurs plus grandes ou égales à r_{i+1} sont ODE si $r_{i+1} - r_i > EXCL_SIGMAGAP$, où la valeur de i la plus petite correspond à la position du point de départ pour la détection vers la droite et, $i+1$ représente la position de l'enregistrement suivant vers la droite. Pour la détection des valeurs aberrantes vers la droite, r_{i+1} ainsi que toutes les valeurs plus grandes ou égales à r_{i+1} sont ODI dès que $r_{i+1} - r_i > IMP_SIGMAGAP$.

Pour la détection des valeurs aberrantes vers la gauche, r_{i-1} ainsi que toutes les valeurs plus petites ou égales à r_{i-1} sont ODE si $r_i - r_{i-1} > EXCL_SIGMAGAP$, où la plus grande valeur de i correspond à la position du point de départ pour la détection vers la gauche et, $i-1$ représente la position de l'enregistrement suivant vers la gauche. Pour la détection des valeurs aberrantes vers la gauche, r_{i-1} ainsi que toutes les valeurs plus petites ou égales à r_{i-1} sont ODI dès que $r_i - r_{i-1} > IMP_SIGMAGAP$.

Si le même écart déclenche la définition des intervalles ODE et ODI en même temps alors, il n'y aura que des valeurs à imputer (ODI). Le système identifie les enregistrements qui ont des valeurs à exclure ou à imputer. Il écrit les statuts FTE et FTI dans le fichier de sortie. Par exemple, si r_8 est ODE, alors x_8 obtiendra le statut FTE dans le fichier de statuts.

Description de la méthode de l'écart-sigma utilisant des tendances historiques

Il s'agit d'un cas particulier de la méthode de l'écart-sigma utilisant des ratios où la variable auxiliaire y_i est la variable x_i choisie d'une période antérieure, c'est-à-dire $x_{i(t-1)}$. Les étapes à suivre avec des tendances historiques sont les mêmes que celles avec des ratios avec la variable choisie représentée par x_{it} et sa valeur historique par $x_{i(t-1)}$, où t et $t-1$ représentent les périodes

de temps. Donc, le ratio pour l'enregistrement i est $r_i = \frac{x_{it}}{x_{i(t-1)}}$.

Exemple d'application de la méthode de l'écart-sigma utilisant des tendances historiques

Les données utilisées pour l'exemple de la méthode Hidiroglou et Berthelot avec des tendances historiques sont utilisées de nouveau ici à titre comparatif. Il s'agit des 24 enregistrements pour deux périodes d'enquête. Les enregistrements x_{23} et x_{24} sont de nouveau ignorés puisqu'ils contiennent une valeur nulle et une valeur négative respectivement. De plus, les valeurs aberrantes seront détectées du côté gauche ainsi que du côté droit (SIDE=BOTH).

En premier lieu, les tendances historiques r_i sont calculées et triées en ordre croissant. L'option STARTCENTILE est définie avec la valeur 75. Tout d'abord, en examinant les données vers la droite, le 75^{ème} centile à partir de l'enregistrement ayant la plus petite valeur est le ratio $r_{17}=1,179$. Le ratio r_{17} servira de point de départ ($r_i = r_{17}$). Le premier écart calculé pour la détection vers la droite ($r_{i+1} - r_i$) est l'écart entre les ratios r_{17} et r_{18} , soit $1,429-1,179=0,250$. Cet écart sera le premier écart vers la droite à être comparé aux écarts-sigma d'exclusion et d'imputation. Le deuxième écart sera $r_{19} - r_{18}$ et ainsi de suite.

Le même procédé est appliqué vers la gauche. Le 75^{ième} centile en partant de la valeur la plus grande et en se dirigeant vers la gauche est le ratio $r_6=0,914$. Ce ratio devient donc le point de départ ($r_i=r_6$) pour la détection vers la gauche. Le premier écart calculé pour la détection vers la gauche ($r_i - r_{i-1}$) est l'écart entre les ratios r_6 et r_5 , $0,914-0,889=0,025$. Cet écart sera le premier écart vers la gauche à être comparé aux écarts-sigma d'exclusion et d'imputation. Le deuxième écart sera $r_5 - r_4$ et ainsi de suite.

L'écart médian absolu calculé qui sera utilisé pour calculer l'écart est de 0,18. Les multiplicateurs pour l'exclusion et pour l'imputation utilisés sont définis respectivement à $\beta_E=1,5$ et $\beta_I=3,0$.

L'écart-sigma pour l'exclusion est calculé et il sera utilisé pour détecter les ODE vers la gauche et vers la droite :

$$EXCL_SIGMAGAP = \beta_E \sigma_{MAD} = (1,5) * (0,18) = 0,27$$

Dès qu'un écart calculé entre deux valeurs successives est plus grand que 0,27, la plus grande de ces deux valeurs (pour la détection vers la droite) ou la plus petite de ces deux valeurs (pour la détection vers la gauche) est identifiée comme valeur à exclure (ODE).

L'écart-sigma pour l'imputation est calculé et il sera utilisé pour détecter les ODI vers la gauche et vers la droite :

$$IMP_SIGMAGAP = \beta_I \sigma_{MAD} = (3,0) * (0,18) = 0,54$$

Dès qu'un écart calculé entre deux valeurs successives est plus grand que 0,54, la plus grande de ces deux valeurs (pour la détection vers la droite) ou la plus petite de ces deux valeurs (pour la détection vers la gauche) est identifiée comme valeur à imputer (ODI).

En d'autres termes, tout écart tombant dans l'intervalle (0,27, 0,54] identifiera les ODE et tout écart plus grand que 0,54 identifiera les ODI. Par exemple, l'écart entre les ratios r_5 et r_6 est 0,025, qui est plus petit que 0,27 alors le ratio $r_5=0,889$ n'est pas un ODE ni un ODI. Cependant, l'écart entre les ratios r_2 et r_3 est $0,781-0,429=0,352$ qui tombe dans l'intervalle (0,27, 0,54] et donc, fait du ratio r_2 un ODE. Le fait que r_2 tombe dans l'intervalle ODE signifie que x_2 obtiendra le statut FTE.

En comparant la méthode de l'écart-sigma avec la méthode Hidiroglou et Berthelot avec le même jeu de données, les valeurs à être imputer (ODI) suivantes selon la méthode Hidiroglou et Berthelot sont maintenant à exclure (ODE) selon la méthode de l'écart-sigma; x_1 et x_2 ont un statut FTE. Vers la droite, les valeurs des trois mêmes enregistrements sont détectées; x_{20} et x_{21} ont toujours un statut FTE et, x_{22} à toujours un statut FTI.

Ordre des valeurs aberrantes dans l'ensemble de données de sortie

L'ensemble de données de sortie de la procédure contient une observation pour chaque champ qui a été identifié comme étant une valeur aberrante à exclure (FTE) ou à imputer (FTI). Chaque observation contient le numéro d'enregistrement, le nom du champ, le statut du champ et plusieurs autres variables. L'utilisateur peut trier ces observations selon l'ordre désiré après avoir exécuté la

Procédure en utilisant la procédure SORT de SAS. Par défaut, lorsqu'on utilise soit la méthode de l'écart-sigma vers la droite soit la méthode Hidiroglou-Berthelot vers n'importe quel côté, toutes les variables associées avec chaque observation sont automatiquement triées à partir de la colonne la plus à droite vers celle la plus à gauche en ordre croissant sauf le numéro d'enregistrement.

Par contre, quand on cherche des valeurs aberrantes à la gauche utilisant la méthode de l'écart-sigma, le premier enregistrement inscrit est celui dont la valeur de son champ a provoqué le premier écart-sigma. Dans ce cas, les valeurs aberrantes qui restent vers la gauche sont subséquemment listées selon la valeur du champ en ordre décroissant.

5. PROC ERRORLOC – LOCALISATION DES ERREURS

Objet

La procédure de localisation des erreurs a pour objet de repérer les champs qui doivent être modifiés dans chaque enregistrement erroné de façon à ce que ce dernier satisfasse à toutes les règles de vérification. Cette procédure ne modifie pas les données initiales: il repère les champs qui doivent faire l'objet d'une imputation, mais il n'effectue aucune imputation. Les valeurs qui remplaceront les valeurs initiales dans ces champs ne sont pas déterminées avant l'étape de l'imputation.

Description de la méthode employée

Les données devant être vérifiées par Banff sont présumées numériques et continues. Dans les règles de vérification, toutes les variables doivent être limitées en dessus et/ou en dessous. Des règles de vérification fondées sur la positivité des valeurs peuvent être ajoutées (avec l'option pour rejeter les valeurs négatives) pour chaque variable aux règles spécifiées par l'utilisateur pour former le système d'inégalités linéaires qui doit être appliqué à chaque enregistrement de données. Les règles spécifiées par l'utilisateur peuvent comprendre des règles fondées sur une égalité ainsi que des règles fondées sur une inégalité. Soit x_1 à x_n les n variables fournies par un répondant à une enquête; on peut alors définir de la façon suivante un système composé de m règles fondées sur une inégalité et/ou une égalité spécifiées par l'utilisateur et de n règles pour limiter les variables :

$$\begin{array}{lllll} a_{11}x_1 + & a_{12}x_2 + \cdots & a_{1n}x_n & \leq & b_1 \\ . & . & \cdots & \leq & . \\ . & . & \cdots & . \leq & . \\ . & . & \cdots & . \leq & . \\ a_{m1}x_1 + & a_{m2}x_2 + \cdots & a_{mn}x_n & \leq & b_m \\ & & & x_1 \geq & 0 \\ & & & . \geq & . \\ & & & . \geq & . \\ & & & x_n \geq & -9999 \end{array}$$

Les règles de vérification fondées sur une inégalité définissent une région, appelée région d'acceptation, dans l'espace à n dimensions. Lorsqu'on substitue les valeurs fournies par le répondant aux variables comprises dans ces règles, tous les enregistrements qui satisfont aux règles se situent à l'intérieur de la région d'acceptation, tandis que ceux qui ne satisfont pas à au moins une règle se situent à l'extérieur de cette région. Certains enregistrements peuvent être rejettés parce que les valeurs visées par la règle sont manquantes ou inconnues, tandis que d'autres peuvent l'être parce qu'ils comportent des valeurs qui ne satisfont pas aux règles de vérification.

Le choix des champs à imputer dans les enregistrements qui ne satisfont pas aux règles de vérification doit reposer sur certains critères. La stratégie employée par Banff consiste à réduire au minimum le nombre de champs devant faire l'objet d'une imputation. Autrement dit, il serait impossible de faire en sorte qu'un enregistrement satisfasse aux règles de vérification en modifiant

un nombre de champs inférieur à celui qui est indiqué dans la solution fournie par la procédure de localisation des erreurs. Cette stratégie constitue une application de la **règle du changement minimal** proposée par Fellegi et Holt (1976) et mise au point par Sande (1979). Elle est intuitivement intéressante parce qu'elle permet de conserver le plus grand nombre possible des données fournies par le répondant. Il y a lieu de noter qu'il y a une différence entre réduire au minimum le nombre de champs à imputer et réduire au minimum l'ampleur de l'imputation. La procédure de localisation des erreurs choisit toujours d'imputer un seul champ, même si ce dernier doit être modifié de beaucoup, plutôt que d'imputer deux champs qui ne nécessitent qu'une modification moins importante.

La procédure de localisation des erreurs a pour objet de repérer les champs qui doivent être imputés afin de « faire entrer » un enregistrement dans la région d'acceptation. La détermination de ces champs s'appelle la solution du problème de localisation des erreurs, alors que le nombre de champs devant faire l'objet d'une imputation s'appelle la **cardinalité** de la solution. Pour en arriver à la solution, certaines « corrections » à chaque enregistrement sont calculées, mais celles-ci ne sont jamais réellement utilisées pour corriger les données. De fait, les « corrections » ne conviendraient pas à cette fin puisqu'elles auraient pour effet de déplacer un enregistrement qui ne satisfait pas aux règles de vérification en un point de la frontière de la région d'acceptation. Aucun utilisateur ne voudrait que tous les rejets à la vérification soient imputés de manière à ce que les enregistrements résultants se situent tout juste aux limites extrêmes d'acceptation.

On trouve ci-après un exposé simplifié des étapes que la procédure de localisation des erreurs suit lorsqu'un enregistrement ne satisfait pas à une ou plusieurs règles de vérification. Le terme « corrections » est employé dans le contexte de la localisation des erreurs; il ne désigne pas l'imputation réelle dont l'enregistrement fera éventuellement l'objet.

- Au début, on ignore quels champs doivent être imputés pour que l'enregistrement puisse satisfaire aux règles de vérification. L'approche adoptée consiste à considérer pour chaque champ des corrections distinctes qui deviennent les variables du problème de localisation des erreurs. Il est probable qu'on finira par constater que bon nombre de ces corrections sont égales à zéro et que les champs correspondants ne nécessitent pas d'imputation, mais on l'ignore au début.
- La procédure impose comme condition que le nombre de corrections non nulles soit réduit au minimum ou, par équivalence, que le plus grand nombre possible de corrections soient fixées à zéro. Cette condition permet de garantir que le nombre de champs à imputer est un minimum.
- Lorsque les corrections sont appliquées aux champs, l'enregistrement résultant doit, par définition, satisfaire aux règles de vérification. Le système substitue donc les valeurs corrigées aux variables correspondantes dans les règles et il résout le système linéaire afin d'obtenir les valeurs des corrections inconnues.

- Un champ n'a pas besoin d'être imputé si le système détermine que la correction correspondante est égale à zéro. Les champs auxquels correspondent des corrections non nulles seront imputés dans des procédures ultérieures de manière à ce que l'enregistrement entre dans la région d'acceptation. Le système désigne donc ces champs comme étant ceux qui doivent être imputés. À cette fin, il leur attribue un code FTI (champ à imputer) dans le fichier de sortie de statuts des champs.

Les solutions multiples

Pour tout enregistrement donné, le problème de localisation des erreurs peut trouver plusieurs solutions nécessitant le repérage du même nombre de champs à imputer. Par exemple, si un enregistrement ne satisfait pas à la règle $x_1 + x_2 + x_3 = x_4$, il importe peu que l'on choisisse d'imputer x_1, x_2, x_3 ou x_4 , pour autant qu'aucune de ces valeurs n'ait été rejetée lors de l'application d'autres règles de vérification. La procédure de localisation des erreurs détermine toutes les solutions possibles qui portent sur le nombre minimal de champs, puis en choisit une au hasard. Il est donc possible que des résultats légèrement différents soient obtenus lorsque le même ensemble de données est soumis deux fois à la procédure de localisation des erreurs, puisque ce dernier peut choisir des solutions différentes pour les enregistrements auxquels s'appliquent des solutions multiples. Il y a lieu de noter que seule la solution choisie est retenue; la procédure de localisation des erreurs écarte toutes les autres solutions.

Exemple de localisation des erreurs

Prenons par exemple l'ensemble de règles de vérification suivant, qui définit la région d'acceptation illustrée à la figure 5.1.

$$x + y \geq 6 \quad (1)$$

$$x \leq 4 \quad (2)$$

$$y \leq 5 \quad (3)$$

Dans le cas de l'enregistrement A, les valeurs pour x et y sont (3,4). Cet enregistrement se situe à l'intérieur de la région et il satisfait à toutes les règles de vérification. Les enregistrements B (2,3), C (4,1) et D (5,6) ne satisfont pas à au moins une règle, mais il est possible de faire en sorte qu'ils satisfassent aux règles en modifiant l'une des valeurs déclarées ou les deux. Pour chaque enregistrement, la procédure de localisation des erreurs détermine le nombre minimal de champs qui doivent être modifiés et désigne ce groupe de champs comme étant la solution pour cet enregistrement.

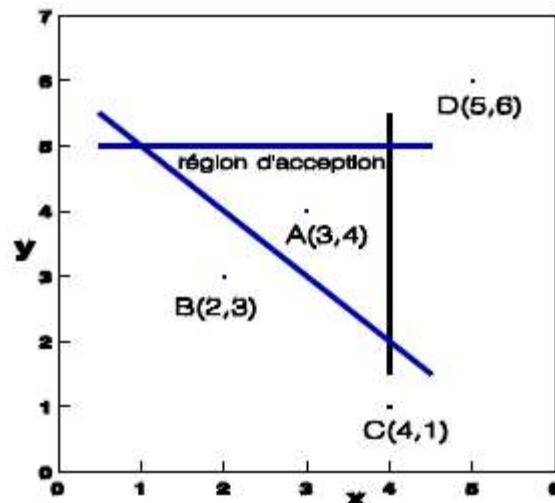


Figure 5.1 Exemple de région d'acceptation

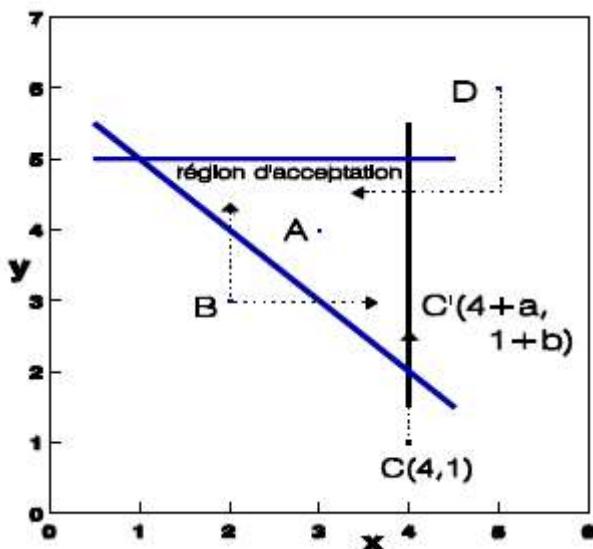


Figure 5.2 Solutions pour les enregistrements A, B et D

également possible de rendre l'enregistrement C acceptable en modifiant tant x que y, mais comme le changement de y entraîne la modification d'un nombre inférieur de champs, la procédure de localisation des erreurs désignerait donc ce changement comme solution. La solution pour cet enregistrement est exposée en détail ci-après.

De même, l'enregistrement D ne satisfait pas aux règles de vérification; pour qu'il entre dans la région d'acceptation, il faudrait imputer tant x que y. Comme l'illustre la figure 5.2, on peut d'abord modifier y, comme l'indique la ligne pointillée verticale, puis modifier x, comme l'indique la ligne pointillée horizontale. Il y a lieu de noter que x et y sont désignés comme étant le groupe de champs à imputer dans l'enregistrement D; la modification nécessaire pourrait aussi être représentée par une ligne horizontale suivie d'une ligne verticale.

Pour illustrer de façon plus détaillée de quelle manière le système s'y prend pour résoudre le problème de localisation des erreurs, prenons l'enregistrement C (4,1) qui se situe à l'extérieur de la région d'acceptation. Dans Banff, les corrections réellement calculées seraient égales à la différence entre deux valeurs non négatives, dont au moins une devrait être égale à zéro. Toutefois, pour simplifier le problème, nous avons choisi d'appliquer des corrections uniques à chaque champ dans l'exemple qui suit.

Au début, on ignore quels sont le ou les champs à imputer. Des corrections inconnues de a et b sont donc considérées pour x et y respectivement, afin de « déplacer » l'enregistrement initial C (4,1) jusqu'à un point inconnu C' (4+a,1+b) situé à l'intérieur de la région d'acceptation. Les coordonnées du point C' doivent satisfaire aux règles de vérification, puisque nous exigeons que C' se situe à l'intérieur de la région d'acceptation; ainsi, en faisant les substitutions appropriées dans les règles fondées sur une inégalité contenues dans l'ensemble de règles de vérification initial, nous obtenons les règles suivantes:

La figure 5.2 représente les solutions possibles au problème de localisation des erreurs pour les enregistrements B, C et D.

Il est possible d'amener l'enregistrement B dans la région d'acceptation en modifiant soit la valeur x, comme l'indique la ligne pointillée horizontale originant en B, soit la valeur y, comme l'indique la ligne pointillée verticale originant en B. Nous sommes donc en présence d'un exemple de solutions multiples, chacune ayant une cardinalité de 1. En pareil cas, Banff choisirait au hasard l'une des deux solutions et laisserait tomber l'autre.

Il est possible d'amener l'enregistrement C dans la région d'acceptation en modifiant la valeur y, comme l'indique la ligne pointillée verticale originant en C. Il serait

$$4 + a + 1 + b \geq 6 \quad (1)$$

$$4 + a \leq 4 \quad (2)$$

$$1 + b \leq 5 \quad (3)$$

ou, plus simplement,

$$a + b \geq 1 \quad (1)$$

$$a \leq 0 \quad (2)$$

$$b \leq 4 \quad (3)$$

Banff exige que le nombre de champs à imputer soit réduit au minimum. Il existe un nombre infini de solutions possibles où $a=0$ et où b se situe entre 1 et 4. Ces solutions équivalent à accepter la valeur de x et à choisir y pour fin d'imputation. Il n'y a aucune solution où $b=0$. Il existe aussi un nombre infini de solutions possibles où tant a que b ne prennent pas une valeur égale à zéro, mais ces solutions nécessiteraient l'imputation de deux champs, tandis qu'il suffirait autrement d'en choisir un seul. Par conséquent, la procédure de localisation des erreurs accepterait la valeur x et désignerait la valeur y comme étant celle devant faire l'objet d'une imputation.

L'algorithme de Chernikova

Le problème de localisation des erreurs est exprimé sous la forme d'un programme linéaire assujetti à une contrainte de cardinalité (Sande, 1979) et est résolu à l'aide de l'algorithme de Chernikova (Chernikova, 1964 et 1965), (Rubin, 1973). Le système construit une matrice à partir des règles de vérification spécifiées par l'utilisateur, des règles de vérification fondées sur la positivité des valeurs s'ils existent, et des valeurs initiales de l'enregistrement. Plusieurs itérations sont d'habitude nécessaires pour en arriver à la solution définitive. À chaque itération, les colonnes de la matrice sont soit retenues, soit prises en combinaison linéaire avec une ou plusieurs autres colonnes. De cette façon, la matrice augmente souvent de beaucoup en taille, même si certaines colonnes sont éliminées parce qu'elles ne peuvent jamais amener à une solution. Le temps d'exécution nécessaire pour en arriver à une solution augmente aussi du fait qu'un plus grand nombre de colonnes sont produites et que les essais de suivi sont effectués plus souvent.

Bien sûr, l'application de l'algorithme de Chernikova à Banff est en réalité un peu plus complexe. Le lecteur souhaitant obtenir une description plus détaillée de cette application est invité à consulter l'ouvrage de Schiopu-Kratina et Kovar (1989).

Règles de vérification pour le traitement des valeurs négatives

La formulation des règles de vérification pour le traitement des valeurs négatives peut poser les défis qui peuvent produire les résultats imprévus si l'utilisateur ne les anticipe pas en avance. Cette situation demande une attention spéciale. Pour plus d'information et des exemples, voyez le document « Spécification des règles de vérification avec des données négatives dans Banff ». (Équipe de soutien de Banff, 2006).

L'utilisation de poids dans la procédure de localisation des erreurs

Dans certains cas, il peut arriver que l'utilisateur souhaite influer sur le choix des champs retenus pour fin d'imputation. À cette fin, il peut élargir la règle du changement minimal afin que cette dernière tienne compte des poids attribués à chaque variable. La procédure de localisation des erreurs minimise ensuite la somme des poids des champs qui sont désignés pour être imputés. Par exemple, si le poids d'une variable est de 2, Banff considère la modification de ce champ comme étant exactement identique à la modification de deux champs dont le poids est de 1. En pareil cas, la cardinalité de la solution est égale à la somme des poids des champs désignés comme devant faire l'objet d'une imputation. Les variables peuvent se voir attribuer des nombres fractionnés comme poids. Si le poids d'une variable est de 1.1 tandis que celui de toutes les autres est de 1, une solution portant sur un champ dont le poids est de 1.1 aura une cardinalité inférieure à celle d'une solution portant sur deux champs, mais une cardinalité supérieure à celle de solutions portant sur tout autre champ unique. Ces poids s'appliquent à tous les enregistrements du groupe de données et du groupe de règles de vérification en cours de traitement; il est impossible d'attribuer des poids différents aux divers enregistrements. En d'autres termes, si l'utilisateur désire soumettre le premier enregistrement de données à la procédure de localisation des erreurs avec un certain ensemble de poids, la localisation des erreurs est alors effectuée sur tout le groupe d'enregistrements à l'aide du même ensemble de poids.

Il faut faire preuve de prudence dans l'attribution des poids, puisque les variables sont habituellement visées par plusieurs règles de vérification avec différentes combinaisons d'autres variables. Le fait d'attribuer, dans une règle de vérification, un poids exceptionnel à une certaine variable peut avoir une incidence sur les poids devant être attribués aux autres variables visées par d'autres règles. L'utilisation de poids peut aussi influer sur le temps d'exécution nécessaire pour que la procédure de localisation des erreurs en arrive à ses solutions. Si la matrice de travail n'atteint pas la taille maximale permise, le temps d'exécution peut être réduit. Par contre, si la matrice de travail devient si grande qu'il est nécessaire d'éliminer certaines colonnes, le temps d'exécution peut augmenter. Lorsque toutes les variables ont des poids égaux, la suppression de toutes les colonnes dont le poids total (ou cardinalité) est le plus élevé entraîne souvent l'élimination de nombreuses colonnes. Toutefois, lorsque les poids sont inégaux, la suppression de toutes les colonnes ayant la cardinalité la plus élevée se traduit souvent par l'élimination de seulement quelques colonnes. De plus, lorsque les colonnes ayant la cardinalité maximale ont été supprimées, une nouvelle cardinalité maximale est calculée en soustrayant .5 du maximum courant. Lorsque les variables ont des poids différents, plusieurs passages en machine peuvent être nécessaires pour réduire la nouvelle cardinalité maximale à un niveau qui permette de limiter la production de nouvelles colonnes afin d'éviter que la taille totale de la matrice ne dépasse la limite de mémoire. Nous décrivons ci-dessous de nombreux exemples d'utilisation de poids.

Il est possible que les valeurs de certaines des variables comprises dans un groupe de règles de vérification soient déclarées de façon plus fiable que d'autres. Ainsi, les répondants peuvent avoir tendance à indiquer de bonnes valeurs pour le revenu brut tiré d'une entreprise, mais à déclarer des

valeurs médiocres pour le total des salaires, ne sachant pas s'ils doivent inclure les primes dans ce total. En pareil cas, les utilisateurs de Banff attribuent souvent un poids plus élevé à la variable la plus fiable pour que, lorsque le système doit choisir entre deux champs, il retienne le plus fiable et soumette l'autre à l'imputation. Il faut de nouveau insister sur le fait que l'ensemble de poids précisé par l'utilisateur s'applique à tous les enregistrements du groupe de données et du groupe de règles de vérification en cours de traitement.

Prenons par exemple une variable pour laquelle on déclare souvent une valeur de zéro et qui fait partie de la règle de vérification suivante fondée sur une égalité :

$$\text{blé} + \text{avoine} + \text{orge} + \text{seigle} + \text{carvi} = \text{total des grains produits.}$$

Comme très peu d'exploitations agricoles produisent des graines de carvi, si l'enregistrement ne satisfait pas à cette règle, il est probable qu'une des autres valeurs a été déclarée incorrectement et qu'il faudrait modifier cette valeur plutôt que de remplacer la valeur zéro de la variable « carvi » par une autre valeur. Toutefois, si les variables ont des poids égaux, la variable « carvi » a autant de chances d'être choisie pour être imputée que n'importe laquelle des quatre autres variables. La solution consiste à attribuer à la variable « carvi » un poids un peu plus élevé qu'aux autres. Il s'agit d'un cas particulier où la valeur d'une variable est déclarée de façon plus fiable que d'autres. L'utilisateur croit que le répondant type est davantage au fait de la quantité de graines de carvi produites que des quantités des autres types de grain, parce que la production de carvi est presque toujours nulle. On peut noter que la variable « carvi » ne serait jamais choisie pour fin d'imputation si la valeur déclarée était nulle et si le total déclaré était inférieur à la somme des valeurs des variables, puisqu'il faudrait alors réduire la valeur d'au moins une des variables et qu'il serait impossible de réduire la valeur nulle tout en satisfaisant aux règles de vérification.

Il est possible d'utiliser des poids pour faire en sorte que certains champs ne puissent jamais être choisis pour fin d'imputation, ce qui peut être utile lorsqu'une variable appartient à au moins deux groupes de règles de vérification. L'utilisateur doit éviter qu'un enregistrement satisfasse aux règles d'un groupe (que ce soit initialement ou après imputation), pour être ensuite imputé dans un deuxième groupe de règles de vérification d'une façon telle qu'il ne satisfasse plus aux règles du premier groupe. On peut éviter ce problème en attribuant à n'importe quelle variable qui fait partie d'un groupe de règles de vérification qui a précédemment fait l'objet d'une imputation un poids plus élevé que la somme des poids de toutes les autres variables. Évidemment, l'ordre de traitement des groupes de règles de vérification revêt une grande importance en pareil cas et il doit être étudié attentivement.

Les poids sont souvent utilisés conjointement avec des règles de vérification fondées sur une égalité afin de rendre certaines variables moins susceptibles d'être choisies pour fin d'imputation lorsqu'un enregistrement ne satisfait pas à une de ces règles. Prenons le cas où la somme des valeurs de quatre variables doit être égale à la valeur d'une cinquième variable. Supposons que l'utilisateur désire conserver la somme pour autant qu'il soit possible de modifier un ou deux éléments de l'équation de façon à satisfaire à l'égalité, mais qu'il préférerait modifier la somme plutôt que d'avoir à imputer trois ou quatre éléments. Il peut arriver à ses fins en attribuant des poids de 1 à chacun des éléments et un poids supérieur à 2 mais inférieur à 3 à la somme. Par contre, si l'utilisateur préfère toujours conserver les éléments et rajuster la somme, il doit alors attribuer à celle-ci un poids inférieur à celui de chacune des quatre autres variables.

Limite du nombre de champs visés par la solution

La procédure de localisation des erreurs comporte une autre caractéristique qui permet à l'utilisateur de limiter le nombre maximal de champs qui sont désignés comme nécessitant une imputation dans toute solution. La même limite s'applique à tous les enregistrements du groupe de données et du groupe de règles de vérification en cours de traitement. Cette limite a pour objet de restreindre le nombre de champs (ou la somme des poids des champs si des poids sont utilisés) à imputer dans la solution définitive. Bien sûr, cette limite n'est efficace que si elle est inférieure au nombre (pondéré) de champs.

Aucune solution trouvée - Imputation manuelle requise

Si l'utilisateur n'a pas limité le nombre (pondéré) de champs à désigner comme devant être imputés, il est toujours possible de trouver une solution. Dans le pire cas, cette solution désigne tous les champs pour fin d'imputation, comme pour l'enregistrement D dans l'exemple utilisé précédemment. Toutefois, si une limite est spécifiée, il se peut qu'il existe des enregistrements pour lesquels il n'y a aucune solution portant sur un nombre (pondéré) de champs qui soit inférieur ou égal à cette limite. Comme la seule solution pour l'enregistrement D portait sur deux champs, aucune solution n'aurait pu être trouvée si la limite du nombre de champs à imputer avait été fixée à un. Lorsque Banff ne peut trouver aucune solution en raison de cette limite, il simplement rejette l'enregistrement entier et il faire notification de cette action à l'utilisateur. Le fait de limiter le nombre de champs visés par la solution a pour avantage de permettre d'éviter que la matrice de travail utilisée dans les calculs n'atteigne la taille qui aurait été nécessaire pour trouver une solution portant sur de nombreux champs. On réduit ainsi le temps d'exécution, mais il est possible qu'un plus grand nombre d'enregistrements nécessitent une forme quelconque d'imputation manuelle parce qu'aucune solution acceptable n'a pu être trouvée.

Aucune solution trouvée - Délai dépassé

L'utilisateur peut limiter le temps d'exécution que le système peut consacrer à la recherche de la solution pour un enregistrement donné. Si la solution de cardinalité minimale n'a pas été trouvée dans le délai indiqué, l'enregistrement entier est rejeté et une notification à l'utilisateur est fait. Lors des passages en machine normaux, il faudrait toujours introduire une valeur dans ce champ pour éviter qu'un très petit nombre d'enregistrements ne nécessite un temps d'exécution très élevé. Il est difficile de recommander une valeur implicite précise puisque le temps nécessaire pour traiter les enregistrements varie grandement d'un groupe de règles de vérification et d'un enregistrement à l'autre. Néanmoins, l'utilisateur pourrait préciser un délai se situant entre 0.5 et 30 secondes pour les travaux exécutés sur l'ordinateur principal, puis modifier ce délai après avoir étudié le nombre d'enregistrements rejetés qui sont produits.

L'utilisateur peut soumettre de nouveau les enregistrements rejetés au traitement après avoir allongé le délai. Il lui incombe de s'assurer que le délai a bien été allongé; Banff ne conserve pas le(s) délai(s) précédemment fixé(s) pour vérifier cette condition. Si le délai n'est pas supérieur à celui qui avait été fixé pour le passage en machine qui a produit les enregistrements, aucun nouvel enregistrement ne sera traité. L'utilisateur peut aussi traiter les enregistrements en-dehors de Banff.

Autres sources de champs à imputer

Il se peut que certains champs d'un enregistrement aient déjà été repérés comme étant des champs devant faire l'objet d'une imputation avant que l'enregistrement ne soit soumis à la procédure de localisation des erreurs. Cette situation se produit lorsque la procédure de détection des valeurs aberrantes a été utilisé pour attribuer des codes FTI à certains champs, ou les champs aient déjà été

identifiés manuellement en-dehors de Banff comme champs à imputer. Se reporter à la section 4 du présent guide pour obtenir de plus amples renseignements sur la procédure de détection des valeurs aberrantes. Afin de s'assurer que ces champs sont identifiés comme champs à imputer par Proc Errorloc, il faut changer les valeurs à valeurs manquantes avant que les données sont soumis à Proc Errorloc.

6. PROC DETERMINISTIC – IMPUTATION DÉTERMINISTE

Objet

La procédure d'imputation déterministe analyse chaque champ que le système a précédemment repéré comme devant faire l'objet d'une imputation pour déterminer s'il n'existe qu'une seule valeur capable de satisfaire aux règles de vérification initiales. Si le système trouve une telle valeur, il l'impute au cours de l'exécution de cette procédure.

Description de la méthode employée

La procédure d'imputation déterministe exécute les étapes suivantes pour chaque enregistrement qui comprend un ou plusieurs champs devant faire l'objet d'une imputation.

- Il élimine les règles de vérification auxquelles satisfont les champs valides de l'enregistrement. Ainsi, toutes les règles qui restent portent sur des champs qui doivent être imputés et, éventuellement, sur des champs acceptables.
- Il substitue les valeurs initiales des variables qui doivent être conservées aux variables comprises dans les règles restantes pour obtenir un ensemble réduit de règles de vérification. Cet ensemble réduit contient les règles qui visent uniquement les champs à imputer.
- Il trouve pour chaque champ les valeurs maximale et minimale pouvant satisfaire à l'ensemble réduit de règles de vérification et, si l'option pour rejeter les valeurs négatives est en vigueur, aux règles fondées sur la positivité des valeurs.
- Si les valeurs maximale et minimale sont identiques, il n'y a qu'une seule valeur possible, et cette dernière est imputée.

On ne peut s'attendre à ce que la procédure d'imputation déterministe trouve une solution acceptable pour la majorité des champs qui ont besoin d'imputation. Néanmoins, il est utile d'exécuter la procédure d'imputation déterministe parce qu'il se peut que les solutions trouvées dans cette procédure ne soient jamais trouvées au moyen des autres méthodes d'imputation. En outre, l'exécution de cette procédure sert à réduire le nombre de champs qui devront être imputés par des procédures ultérieurs. En règle générale, la procédure d'imputation déterministe devrait trouver davantage de solutions si certaines des règles de vérification sont fondées sur une égalité. Toutefois, comme l'illustre l'exemple qui suit, il est possible qu'il existe des solutions déterministes même en l'absence de règles fondées sur une égalité.

Exemple d'imputation déterministe

Prenons par exemple l'ensemble de règles de vérification suivant où l'option de rejeter les valeurs négatives est en vigueur, et les valeurs contenues dans l'enregistrement après la localisation des erreurs.

Règles initiales		Règles de positivité		Enregistrement
$x_1 + x_2 \leq x_3$	(1)	$x_1 \geq 0$	(5)	$x_1 = ?$ (imputation requise)
$.54x_3 + x_4 \leq .9x_1$	(2)	$x_2 \geq 0$	(6)	$x_2 = 400$
$.6x_3 \leq x_1$	(3)	$x_3 \geq 0$	(7)	$x_3 = 1000$
$x_3 \leq 1500$	(4)	$x_4 \geq 0$	(8)	$x_4 = ?$ (imputation requise)

Les règles (1), (2), (3), (5) et (8) portent sur les champs qui doivent être imputés; l'enregistrement ne peut donc pas satisfaire à ces règles pour l'instant. Toutefois, les règles (4), (6) et (7) portent uniquement sur des variables qui doivent être conservées et les valeurs acceptables de l'enregistrement satisfont à ces règles. Par conséquent, la première étape en vue de trouver l'ensemble réduit de règles consiste à éliminer ces trois règles.

Ensuite, le système substitue les valeurs des variables qui doivent être conservées aux variables comprises dans les cinq règles restantes pour obtenir l'ensemble réduit de règles de vérification.

Ensemble réduit de règles			
$x_1 \leq 600$	(1)	$x_1 \geq 0$	(5)
$540 + x_4 \leq .9x_1$	(2)	$x_4 \geq 0$	(8)
$600 \leq x_1$	(3)		

La valeur maximale et la valeur minimale que x_1 peut prendre tout en continuant de satisfaire aux règles de vérification est 600. Cette valeur est donc imputée pour x_1 . La valeur de 600 pour x_1 satisfait aux règles (1), (3), et (5). Ces règles peuvent être éliminées. Puis, les valeurs des variables gardées sont implantées dans les deux règles qui restent. Finalement, nous obtenons l'ensemble réduit de règles.

Ensemble réduit de règles			
$x_4 \leq 0$	(2)	$x_4 \geq 0$	(8)

La seule valeur qui peut satisfaire aux deux règles est zéro, donc la valeur zéro est imputée pour la variable x_4 .

Considérons maintenant le même ensemble de règles original quand l'option pour accepter les valeurs négatives est en vigueur, et les valeurs dans l'enregistrement après la localisation des erreurs.

Règles initiales		Enregistrement
$x_1 + x_2 \leq x_3$	(1)	$x_1 = ?$ (imputation requise)
$.54x_3 + x_4 \leq .9x_1$	(2)	$x_2 = 400$
$.6x_3 \leq x_1$	(3)	$x_3 = 1000$
$x_3 \leq 1500$	(4)	$x_4 = ?$ (imputation requise)

Encore une fois, la première étape enfin de trouver l'ensemble réduit de règles est d'éliminer la

règle (4). Ensuite, le système substitue les valeurs des variables qui doivent être conservées aux variables comprises dans les trois règles restantes pour obtenir l'ensemble réduit de règles de vérification.

Ensemble réduit de règles

$$x_1 \leq 600 \quad (1)$$

$$540 + x_4 \leq .9 x_1 \quad (2)$$

$$600 \leq x_1 \quad (3)$$

La valeur maximale et la valeur minimale que x_1 peut prendre tout en continuant de satisfaire aux règles de vérification est 600. Cette valeur est donc imputée pour x_1 . Dans ce cas-ci, la procédure d'imputation déterministe ne peut trouver aucune solution pour x_4 . Il faut alors utiliser une autre méthode d'imputation.

Règles de vérification pour le traitement des valeurs négatives

La formulation des règles de vérification pour le traitement des valeurs négatives peut poser les défis qui peuvent produire les résultats imprévus si l'utilisateur ne les anticipe pas en avance. Cette situation demande une attention spéciale. Pour plus d'information et des exemples, voyez le document « Spécification des règles de vérification avec des données négatives dans Banff ». (Équipe de soutien de Banff, 2006).

7. PROC DONORIMPUTATION – IMPUTATION PAR ENREGISTREMENT DONNEUR

Objet

La procédure d'imputation par enregistrement donneur emploie la méthode du voisin le plus proche pour trouver, pour chaque enregistrement devant faire l'objet d'une imputation, l'enregistrement valide qui lui ressemble le plus et qui permettra à l'enregistrement receveur imputé de satisfaire aux règles de vérification post-imputation spécifiées par l'utilisateur. L'imputation est exécutée si le système trouve un tel enregistrement. L'imputation par enregistrement donneur est la méthode d'imputation préconisée pour de nombreuses applications parce que tous les champs devant être imputés sont tirés du même enregistrement donneur et que, par conséquent, les rapports entre les variables imputées sont conservés.

Définitions

Par **enregistrement receveur**, on entend un enregistrement dont au moins un des champs doit faire l'objet d'une imputation et fait partie du groupe de règles de vérification en cours de traitement. Ces enregistrements sont repérés par la procédure de localisation des erreurs, ou une autre méthode en dehors de Banff. Il se peut que la procédure de détection des valeurs aberrantes ait été exécuté avant la procédure de localisation des erreurs et qu'il ait lui aussi repéré des champs devant faire l'objet d'une imputation.

Par **enregistrement donneur**, on entend un enregistrement qui satisfait à toutes les règles de vérification et qui ne comporte aucun champ à imputer faisant partie du groupe de règles de vérification en cours de traitement ou des champs d'appariement spécifiés par l'utilisateur à l'extérieur du groupe de règles de vérification. Tous les champs des enregistrements donneurs doivent comprendre des données initialement fournies par les répondants ou des données imputées par la procédure d'imputation déterministe, mais l'utilisateur peut spécifier que les autres données déjà imputées sont acceptable. Les valeurs qui sont imputées par la procédure d'imputation par enregistrement donneur sont tirées directement des enregistrements donneurs. Les enregistrements donneurs peuvent comporter des champs qui ont été désignés comme étant des champs renfermant des valeurs aberrantes à exclure (FTE) et qui, par conséquent, ne peuvent pas servir pour l'imputation par enregistrement donneur.

Par **enregistrement mixte**, on entend un enregistrement qui possède certaines des caractéristiques des enregistrements donneurs et certaines des caractéristiques des enregistrements receveurs. Un tel enregistrement ne renferme aucun champ à imputer faisant partie du groupe de règles de vérification en cours de traitement, mais il n'est pas un enregistrement donneur parce qu'il comporte au moins un champ à imputer faisant partie des champs d'appariement spécifiés par l'utilisateur à l'extérieur du groupe de règles de vérification. De même, il n'est pas un enregistrement receveur, puisqu'il ne comporte aucun champ à imputer faisant partie du groupe de règles de vérification en cours de traitement.

Les codes **FTE (Field to Exclude - valeur aberrante à exclure)** peuvent être attribués à des champs faisant partie de n'importe quel type d'enregistrement. L'utilisateur peut désigner n'importe quel champ comme FTE. Aussi, la procédure de détection des valeurs aberrantes a déterminé que ces champs comportaient des valeurs qui ne sont pas suffisamment extrêmes pour

nécessiter une imputation, mais dont le caractère assez exceptionnel fait qu'elles ne doivent pas être transférées dans d'autres enregistrements. Ces valeurs sont acceptées dans l'enregistrement initial parce qu'elles ont été fournies par le répondant. Ces champs peuvent servir de champs d'appariement lorsqu'ils se trouvent dans un enregistrement donneur. Les champs FTE n'ont aucune incidence dans les enregistrements receveurs.

Les valeurs des **champs d'appariement** sont calculées individuellement pour chaque enregistrement receveur. Il n'est pas nécessaire que les valeurs des champs d'appariement d'un enregistrement receveur correspondent exactement à celles des champs d'appariement de l'enregistrement donneur qui est le voisin le plus proche. Les valeurs contenues dans ces champs d'appariement servent à déterminer la distance entre les enregistrements receveurs et les enregistrements donneurs. Comme la recherche du voisin le plus proche s'effectue à partir de ces champs, il doit exister une relation entre ces derniers et les champs qui doivent être imputés. Il se peut qu'un enregistrement ne comporte aucun champ d'appariement.

Un **champ d'appariement spécifié par l'utilisateur** ou un **champ d'appariement obligatoire** est un champ qui est désigné par l'utilisateur comme champ d'appariement pour tous les enregistrements receveurs, peu importe si le système l'aurait autrement choisi en tant que tel. Les champs de ce genre sont utilisés comme champs d'appariement pour tous les enregistrements receveurs, sauf dans les cas où le champ d'appariement spécifié par l'utilisateur dans l'enregistrement receveur lui-même doit faire l'objet d'une imputation. Comme dans le cas des autres champs d'appariement, il n'est pas nécessaire que la valeur d'un champ d'appariement obligatoire soit identique à celle du champ correspondant dans l'enregistrement donneur qui est le voisin le plus proche.

Les **règles de vérification post-imputation** forment un système d'inégalités linéaires spécifiées par l'utilisateur auxquelles doivent satisfaire les enregistrements imputés dans la présente procédure. Aucune contrainte n'est imposée sur le rapport entre les règles post-imputation et les règles initiales, si ce n'est qu'elles doivent porter sur les mêmes champs. Il est courant de choisir comme règles de vérification postimputation une version moins « stricte » des règles initiales, de sorte que la région d'acceptation résultante englobe entièrement la région définie par les règles initiales. Les égalités contenues dans l'ensemble de règles initiales sont souvent remplacées par deux inégalités afin de permettre un certain écart par rapport à l'égalité stricte et d'augmenter ainsi les chances de réussite de l'imputation. En revanche, on peut rendre l'ensemble de règles de vérification post-imputation plus restrictif que l'ensemble de règles initiales en y ajoutant des règles afin de faire en sorte que certaines conditions, comme l'existence d'une correspondance exacte pour une variable de classification, soient imposées aux données imputées.

Exemple de classement des enregistrements

Dans l'exemple qui suit, le groupe de règles de vérification se compose des variables x_1 à x_4 alors que x_5 est un champ d'appariement spécifié par l'utilisateur ne faisant pas partie du groupe de règles de vérification.

		À l'intérieur du groupe de règles de vérification				À l'extérieur du groupe de règles
	Ident.	x ₁	x ₂	x ₃	x ₄	x ₅
donneur donneur	1	ok	ok	ok	ok	ok
	2	ok	ok	ok	ok	imputé
receveur receveur receveur receveur	3	FTI	appar.-système	appar.- utilisateur	ok	appar.- utilisateur
	4	FTI	appar.- système	appar.- utilisateur	ok	FTI
	5	FTI	appar.- système	FTI	ok	appar.- utilisateur
	6	FTI	FTI	appar.- utilisateur	ok	appar.-utilisateur
mixte	7	ok	ok	ok	ok	FTI

Les enregistrements 1 et 2 sont des enregistrements donneurs. L'enregistrement 1 comporte des valeurs initiales correctes dans tous les champs examinés. Ces valeurs sont désignées comme « ok » dans le tableau ci-dessus. Pour ce qui est de l'enregistrement 2, la valeur de x₅ a été imputée lors de l'exécution antérieure d'un des procédures d'imputation. L'enregistrement 2 est classé parmi les enregistrements donneurs et la valeur imputée est utilisée dans le calcul des distances, bien qu'elle ne pourrait pas être utilisée pour l'imputation par enregistrement donneur parce qu'elle ne fait pas partie du groupe de règles de vérification.

Les enregistrements 3, 4, 5 et 6 sont des enregistrements receveurs parce que chacun d'eux comporte au moins un champ à imputer (FTI) faisant partie du groupe de règles de vérification. Le calcul du voisin le plus proche s'effectuera à partir d'un groupe de champs d'appariement qui peut être différent pour chaque enregistrement receveur. L'utilisateur a choisi deux champs d'appariement : x₃, qui fait partie du groupe de règles de vérification, et x₅ qui n'en fait pas partie. Ces champs sont désignés ci-dessus comme « appar.-utilisateur » lorsqu'ils contiennent des valeurs acceptables. La variable x₃ est utilisée comme champ d'appariement dans tous les enregistrements sauf l'enregistrement 5, où elle ne peut pas l'être parce qu'elle doit faire l'objet d'une imputation; la variable x₅ est utilisée comme champ d'appariement dans tous les enregistrements sauf l'enregistrement 4, où elle ne peut pas l'être parce qu'elle doit faire l'objet d'une imputation.

En plus des champs d'appariement spécifiés par l'utilisateur, Banff choisit des champs d'appariement au moyen d'un algorithme décrit à la sous-section 7.2. Dans l'exemple ci-dessus, les champs d'appariement choisis par le système sont désignés comme « appar.-système ». Le système choisit x₂ comme champ d'appariement pour les enregistrements 3, 4 et 5, mais il ne peut pas choisir cette variable pour l'enregistrement 6, où elle doit faire l'objet d'une imputation. Il est possible qu'un champ soit à la fois un champ d'appariement spécifié par l'utilisateur et un champ d'appariement choisi par le système. Cette possibilité n'a aucune incidence sur le présent exemple parce que, le cas échéant, le classement des enregistrements dans les catégories « donneur », « receveur » et « mixte » resterait le même. Toutefois, Banff écrit les codes à un fichier de sortie SAS pour indiquer si un champ d'appariement a été choisi par l'utilisateur, par le système ou par les deux.

L'enregistrement 7 est un enregistrement mixte. Il n'est pas un enregistrement donneur parce que la

valeur figurant dans le champ x_5 doit être imputée, et il n'est pas un enregistrement receveur parce qu'il ne comporte aucun champ devant être imputé et faisant partie du groupe de règles de vérification.

Dans Banff, l'exécution de la procédure d'imputation par enregistrement donneur se déroule en quatre étapes, lesquelles sont décrites dans les sous-sections suivantes :

- 7.1 Préparation de l'imputation par enregistrement donneur
- 7.2 Détermination des champs d'appariement
- 7.3 Transformation des champs d'appariement
- 7.4 Exécution de l'imputation par enregistrement donneur

Règles de vérification pour le traitement des valeurs négatives

La formulation des règles de vérification pour le traitement des valeurs négatives peut poser les défis qui peuvent produire les résultats imprévus si l'utilisateur ne les anticipe pas en avance. Cette situation demande une attention spéciale. Pour plus d'information et des exemples, voyez le document « Spécification des règles de vérification avec des données négatives dans Banff » (Équipe de soutien de Banff, 2006).

Imputation massive

Certaines enquêtes ont pour échantillon une collection d'unités desquelles un ensemble de données fondamentales est recueilli. Des données plus détaillées ne sont recueillies que pour un sous-échantillon de ces unités. Puis, on peut appliquer la procédure « imputation massive » pour imputer les données plus détaillées aux unités qui ne sont pas dans le sous-échantillon. Nous pouvons considérer l'imputation massive comme un cas particulier de l'imputation par enregistrement donneur, dans lequel les donneurs sont sélectionnés seulement selon quelques champs d'appariement spécifiés par l'utilisateur. On décrit cette procédure à la section 10.

7.1 PRÉPARATION DE L'IMPUTATION PAR ENREGISTREMENT DONNEUR

Objet

L'utilisateur doit spécifier plusieurs paramètres pour que les résultats de l'imputation par enregistrement donneurs répondent à ces besoins. L'utilisateur peut exclure certains enregistrements de la population d'enregistrements donneurs et définir des critères devant être satisfaits pour que le système puisse procéder à l'imputation par enregistrement donneur.

Données précédemment imputées dans des enregistrements donneurs

L'utilisateur peut indiquer si des enregistrements contenant des valeurs précédemment imputées dans au moins un champ du groupe de règles de vérification constituent ou non des enregistrements donneurs admissibles. Toutefois, on doit noter que les valeurs imputées par l'imputation déterministe sont traitées comme des données initiales du répondant. Quoi qu'il en soit, la procédure Proc Donorimputation utilise toujours l'enregistrement donneur le plus rapproché disponible dans l'ensemble choisi, ce qui permet à l'enregistrement résultant de satisfaire les règles de vérification.

Si on conserve les enregistrements contenant les données initiales et les données précédemment imputées comme donneurs, on obtient un bassin d'enregistrements donneurs plus large, et par conséquent cela pourrait produire à trouver un enregistrement donneur approprié pour un plus grand pourcentage d'enregistrements receveurs. Comme la majeure partie des enregistrements sont inclus dans les enregistrements donneurs, ce choix peut mieux représenter la distribution des enregistrements dans l'échantillon. Un désavantage toutefois est que les champs qui ont servi pour l'imputation courrent le risque d'être utilisés de nouveau parce ces champs sont répétés – en effet, on les trouve dans les enregistrements initiaux, et de nouveau dans les enregistrements précédemment imputés.

Si on choisit de ne pas garder les enregistrements donneurs avec les valeurs précédemment imputées, cela limite les enregistrements donneurs potentiels aux enregistrements initialement sans erreur, et donc assure que seulement les données initiales peuvent être utilisées pour l'imputation. Certains utilisateurs estiment qu'il s'agit là d'un avantage important; d'autres préfèrent inclure autant d'enregistrements donneurs que possible dans la population d'enregistrements donneurs. Cependant si on réduit le nombre d'enregistrements donneurs, l'imputation risque de ne pas avoir lieu s'il n'y a pas d'enregistrements donneurs appropriés qui sont disponibles. Quand l'imputation est réussie, l'enregistrement donneur initial le plus rapproché est utilisé, même s'il y avait des enregistrements épurés qui sont plus rapprochés de l'enregistrement receveur que les enregistrements donneurs initiaux.

Autres exclusions de la population d'enregistrements donneurs

Banff identifie les enregistrements donneurs admissibles parmi ceux qui n'ont pas de champ nécessitant une imputation dans le groupe de règles de vérification à traiter ou dans les champs d'appariement spécifiés par l'utilisateur à l'extérieur du groupe de règles de vérification. Toutefois, l'utilisateur peut exclure tout enregistrement de la population des enregistrements donneurs par une identification appropriée dans l'ensemble des données de départ.

L'exclusion de certains types d'enregistrement risque de limiter la population d'enregistrements donneurs. L'utilisateur doit contrebalancer d'une part les avantages découlant de l'exclusion de certains types d'enregistrements inappropriés de la population d'enregistrements donneurs, et

d'autre part le risque de réduire cette dernière à un point tel que le système ne peut pas trouver d'enregistrements donneurs pour bon nombre des enregistrements receveurs.

Limiter l'utilisation d'un enregistrement donneur

En plus de pouvoir exclure certains enregistrements donneurs du processus d'imputation, il est aussi possible de limiter le nombre de fois qu'un enregistrement donneur est utilisé en fournissant au moins un des deux paramètres suivants : *NLIMIT* (nombre limite) et *MRL* (multiplicateur du ratio limite) qui sont utilisés dans le calcul de la limite *DONORLIMIT*, pour chaque groupe *j* de données, comme suit :

$$DONORLIMIT_j = \max \left\{ NLIMIT, \left\lceil MRL * \left\lceil \frac{\# \text{enregistrements receveurs}_j}{\# \text{enregistrements donneurs}_j} \right\rceil \right\rceil \right\}$$

Le symbole $\lceil \rceil$ signifie d'arrondir au plus proche entier vers le haut. Le maximum entre *NLIMIT* et l'autre statistique qui utilise *MRL* est utilisé pour chaque groupe *j* de données en tant que limite appelée *DONORLIMIT_j*. Si un seul paramètre est fourni, *NLIMIT* ou *MRL*, alors l'autre paramètre est exclu dans le calcul de la limite *DONORLIMIT*. Si aucun des deux paramètres n'est spécifié alors, il n'y a pas de limite d'utilisation pour tous les enregistrements donneurs.

Ni le nombre minimum d'enregistrements donneurs requis (paramètre *MINDONORS*) ni le pourcentage minimum d'enregistrements donneurs requis (paramètre *PCENTDONORS*) ne sont affectés par *DONORLIMIT* puisqu'ils sont tous les deux calculés au début de la procédure d'imputation par enregistrement donneur.

Un enregistrement donneur qui a atteint la limite *DONORLIMIT* sera retiré du bassin d'enregistrements donneurs. Il sera ignoré et ne comptera plus dans le nombre maximum d'enregistrements donneurs à essayer (paramètre *N*).

Le processus d'imputation s'arrêtera pour un groupe *j* de données lorsque tous les enregistrements donneurs auront atteint la limite *DONORLIMIT_j*. Dans ce cas, il n'y aura plus de tentative pour trouver un enregistrement donneur pour les enregistrements receveurs qui n'auront pas encore été imputés. En utilisant $MRL \geq 1$, au moins un enregistrement donneur sera disponible pour chaque enregistrement receveur à être imputé.

Critères pour l'imputation par enregistrement donneur

L'utilisateur peut aussi spécifier un pourcentage et un nombre d'enregistrements donneurs dont le système doit disposer pour pouvoir procéder à l'imputation. Si l'expression d'exclusion indiquée risque de réduire considérablement la population d'enregistrements donneurs, l'utilisateur peut utiliser ces critères pour s'assurer que le système dispose d'un pourcentage et d'un nombre minimum d'enregistrements donneurs. Si ces critères ne sont pas satisfait, un message sera imprimé et l'imputation par enregistrement donneur ne sera pas effectuée.

7.2 DÉTERMINATION DES CHAMPS D'APPARIEMENT

Objet

La fonction de détermination des champs d'appariement de Proc Donorimputation analyse chaque enregistrement receveur afin de déterminer un ensemble de champs devant servir au calcul de la distance entre cet enregistrement receveur et les enregistrements donneurs. Il se peut qu'un enregistrement receveur ne comporte aucun champ d'appariement. Lorsque des champs d'appariement existent, ils doivent contenir une partie ou la totalité des valeurs acceptables contenues dans l'enregistrement receveur. Le système trouve habituellement de nombreuses combinaisons différentes de champs d'appariement pour chaque groupe de règles de vérification et chaque groupe de données. Le choix des champs d'appariement est fonction du type de champs à imputer contenus dans chaque enregistrement receveur, des valeurs figurant dans les champs qui sont conservés et, bien sûr, des règles de vérification initiales. Si l'utilisateur a spécifié des champs d'appariement, ceux-ci sont inclus dans l'ensemble de champs d'appariement de l'enregistrement receveur.

Champs d'appariement spécifiés par l'utilisateur ou champs d'appariement obligatoires

Les champs d'appariement spécifiés par l'utilisateur peuvent être soit à l'intérieur, soit à l'extérieur du groupe de règles de vérification. Ils sont utilisés dans le calcul des distances, avec les autres champs d'appariement. Ces champs sont parfois appelés champs d'appariement obligatoires, bien qu'il ne soit pas nécessaire que les valeurs qu'ils contiennent correspondent exactement aux valeurs des champs avec lesquels ils sont appariés.

Les champs d'appariement spécifiés par l'utilisateur provenant de l'extérieur du groupe de règles de

vérification sont particulièrement utiles lorsqu'on croit qu'une variable est en corrélation avec les variables du groupe de règles de vérification, sans apparaître explicitement dans ces règles. Supposons par exemple que le revenu brut tiré d'une entreprise (RBE) ne figure pas dans le groupe de règles de vérification contenant les données sur le nombre d'employés et sur les salaires. Bien qu'aucune relation directe n'ait été précisée, l'utilisateur peut juger que la valeur du RBE contenue dans l'enregistrement receveur devrait influer sur le choix de l'enregistrement donneur correspondant. En pareil cas, l'utilisateur choisirait le RBE comme champ d'appariement provenant de l'extérieur du groupe de règles de vérification.

L'utilisateur devrait vérifier les données pour tous les champs d'appariement potentiels, soit par la procédure de la localisation des erreurs de Banff soit un autre processus. Si non, le calcul des enregistrements voisins les plus proches pourrait être fondé sur des valeurs qui, en fin de compte, seront modifiées. C'est vrai surtout quand les champs d'appariement viennent en-dehors du groupe de règles de vérification. Il incombe à l'utilisateur de déterminer l'ordre de traitement de manière à ce qu'aucun problème de ce genre ne se pose. Si l'imputation et la localisation des erreurs ont été effectuées, les valeurs contenues dans les champs (de l'enregistrement donneur) correspondant aux champs d'appariement spécifiés par l'utilisateur à l'extérieur du groupe de règles de vérification peuvent être des valeurs « acceptables » initiales ou des valeurs qui ont déjà fait l'objet d'une imputation. Dans l'un ou l'autre cas, la valeur contenue dans l'enregistrement donneur est utilisée pour calculer les distances, mais elle ne serait jamais transférée dans un enregistrement receveur parce que seuls les champs se trouvant à l'intérieur du groupe de règles de vérification sont imputés.

L'utilisateur peut aussi choisir des champs se trouvant à l'intérieur du groupe de règles de vérification comme champs d'appariement. Il y a lieu de souligner que de tels champs sont inclus comme champs d'appariement pour tous les enregistrements receveurs qui sont traités lors de ce

passage en machine, même si le système ne les aurait pas autrement choisis comme champs d'appariement. Le seul cas où un champ d'appariement spécifié par l'utilisateur n'est pas utilisé est celui où ce champ doit faire l'objet d'une imputation dans l'enregistrement receveur, ce qui fait qu'il n'y a aucune valeur acceptable à utiliser pour le calcul des distances.

Description de la méthode employée

La procédure exécute chacune des étapes suivantes pour chaque enregistrement receveur.

- Il substitue les valeurs connues acceptables contenues dans l'enregistrement receveur aux variables comprises dans les règles de vérification initiales, puis il exclut les règles qui n'expriment plus que des relations entre des constantes, c'est-à-dire qui ne contiennent plus de variables.
- Les autres règles forment un ensemble réduit, dans lequel les variables à imputer sont les seules inconnues. Cet ensemble de règles de vérification définit une région d'acceptation qui renferme toutes les valeurs acceptables que peuvent contenir les champs à imputer. Le système choisit les règles qui limitent ces champs en déterminant la frontière de cette région d'acceptation. Comme toutes les autres règles faisant partie de l'ensemble de règles réduit sont redondantes, le système les laisse tomber.
- Le système exprime ensuite sous leur forme initiale les règles de vérification qu'il a identifiées lors de l'étape précédente comme faisant partie de la frontière de la région d'acceptation définie par l'ensemble de règles réduit. Il choisit comme champs d'appariement les variables comprises dans ces règles qui ne sont pas des champs devant faire l'objet d'une imputation.

Exemple de détermination des champs d'appariement

Règles initiales

$$x \geq y \quad (1) \quad x \geq 0 \quad (5)$$

$$x \leq 5 \quad (2) \quad y \geq 0 \quad (6)$$

$$y \geq u \quad (3) \quad u \geq 0 \quad (7)$$

$$y \leq 2v \quad (4) \quad v \geq 0 \quad (8)$$

Dans les enregistrements 1 et 2, les valeurs de u et de v sont acceptables, mais les valeurs de x et de y doivent être imputées.

Enregistrement 1

$x = ?$ (imputer)

$y = ?$ (imputer)

$u = 1$

$v = 2$

Enregistrement 2

$x = ?$ (imputer)

$y = ?$ (imputer)

$u = 1$

$v = 3$

Substituons les valeurs acceptables contenues dans les deux enregistrements pour obtenir les relations suivantes.

Enregistrement 1		Enregistrement 2	
$x \geq y$ (1)	$x \geq 0$ (5)	$x \geq y$ (1)	$x \geq 0$ (5)
$x \leq 5$ (2)	$y \geq 0$ (6)	$x \leq 5$ (2)	$y \geq 0$ (6)
$y \geq 1$ (3)	$1 \geq 0$ (7)	$y \geq 1$ (3)	$1 \geq 0$ (7)
$y \leq 4$ (4)	$2 \geq 0$ (8)	$y \leq 6$ (4)	$3 \geq 0$ (8)

Dans le cas des deux enregistrements, les inégalités (7) et (8) expriment uniquement des relations entre des constantes; elles sont donc exclues. Pour ce qui est de l'enregistrement 1, les règles (1) à (6) forment l'ensemble réduit de règles de vérification, les règles (1) à (4) définissant la frontière de la région d'acceptation illustrée dans la première partie de la figure 7.1. En se reportant aux règles initiales, on peut voir que les règles (1) à (4) comprennent les variables x et y , qui doivent être imputées, et les variables u et v , dont les valeurs ont été trouvées acceptables par la procédure de localisation des erreurs. La procédure de détermination des champs d'appariement choisit alors u et v comme champs d'appariement pour l'enregistrement 1.

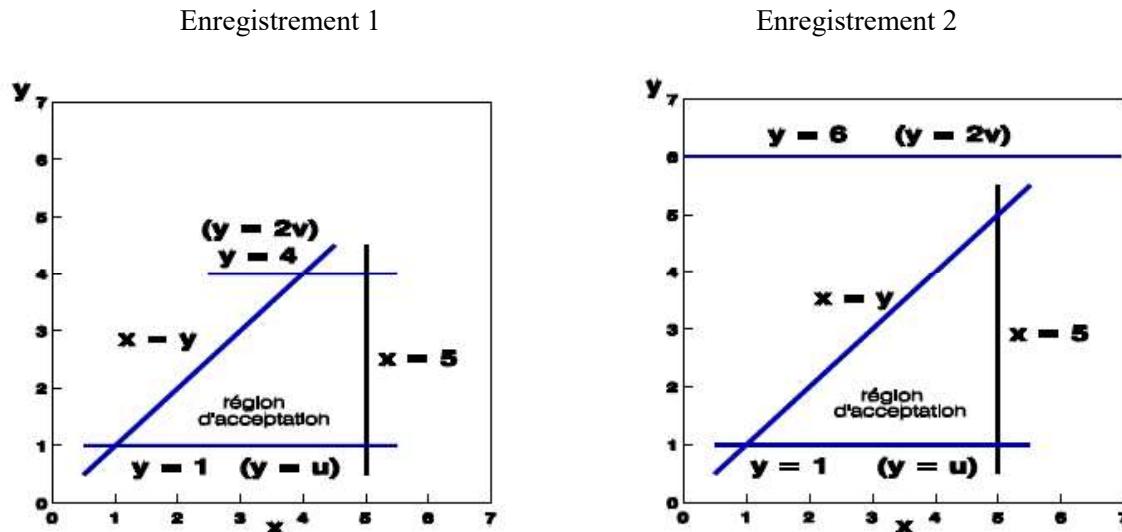


Figure 7.1 Régions d'acceptation définies par l'ensemble réduit de règles de vérification – Enregistrements 1 et 2

En ce qui concerne l'enregistrement 2, les règles (1) à (6) forment aussi l'ensemble réduit de règles de vérification. Dans ce cas, la frontière de la région d'acceptation est définie par les règles (1) à (3), comme il est illustré dans la seconde partie de la figure 7.1. Les règles de vérification initiales (1) à (3) comprennent les variables x et y , qui doivent être imputées, et la variable u , dont la valeur est acceptable et qui est choisie comme champ d'appariement pour l'enregistrement 2.

Les enregistrements 1 et 2 sont visés par les mêmes règles de vérification, ils comportent les mêmes champs à imputer et les valeurs comprises dans leurs champs acceptables sont très similaires. Pourtant, leurs champs d'appariement sont différents. Cette situation s'explique du fait que la valeur de v limite la valeur de y dans l'enregistrement 1, mais qu'elle n'influe pas sur les valeurs de x ou de y dans l'enregistrement 2.

Exemple d'un cas où il n'y a pas de champ d'appariement

Examinons maintenant l'ensemble suivant de règles de vérification et les valeurs contenues dans l'enregistrement 3.

Règles initiales	Enregistrement 3
$x \geq 2$ (1)	$x = ?$ (imputer)
$x \leq 5$ (2)	$y = ?$ (imputer)
$y \geq 1$ (3)	$u = 3$
$y \leq 4$ (4)	$v = 4$
$u + v \leq 10$ (5)	

Substituons dans les règles les valeurs acceptables contenues dans l'enregistrement 3 pour obtenir les relations suivantes.

$$\begin{aligned}x &\geq 2 \quad (1) \\x &\leq 5 \quad (2) \\y &\geq 1 \quad (3) \\y &\leq 4 \quad (4) \\7 &\leq 10 \quad (5)\end{aligned}$$

La règle (5) est exclue parce qu'elle exprime uniquement des relations entre des constantes. Les autres règles, soit les règles (1), (2), (3), et (4), forment l'ensemble réduit de règles de vérification. La région d'acceptation définie par ce dernier est illustrée à la figure 7.2. La frontière de cette région d'acceptation est définie par les règles (1) à (4).

La prochaine étape consiste à examiner les règles (1) à (4) correspondantes dans l'ensemble de règles initiales. On peut voir que les règles (1) à (4) ne comprennent que les variables x et y , qui sont deux champs à imputer. Aucun champ acceptable n'étant visé par ces règles, le système n'a identifié aucun champ d'appariement pour cet enregistrement. Cela équivaut à dire que u et v , les champs acceptables dans l'enregistrement 3, ne fournissent pas de renseignements sur les valeurs qui devraient être imputées pour x et y .

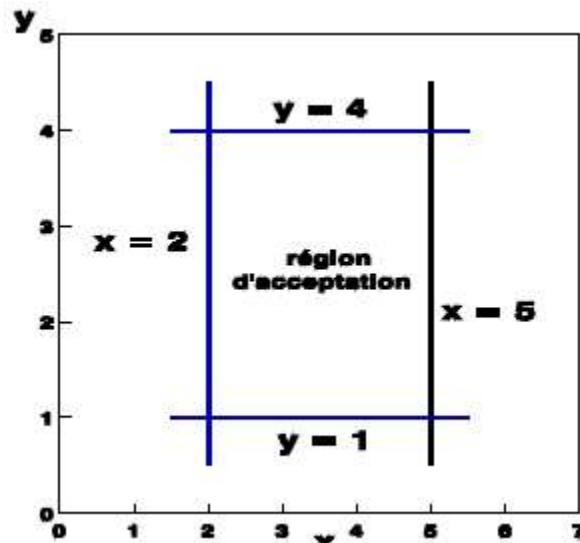


Figure 7.2 Région d'acceptation définie par l'ensemble réduit de règles de vérification – Enregistrement 3

7.3 TRANSFORMATION DES CHAMPS D'APPARIEMENT

Objet

La fonction de transformation des champs d'appariement de Proc Donorimputation effectue une transformation uniforme par rang de toutes les valeurs acceptables contenues dans un groupe de règles de vérification et un groupe de données pour chaque variable qui a été choisie au moins une fois comme champ d'appariement. Cette transformation a pour but d'éliminer l'effet d'échelle des données qui sont utilisées pour le calcul de la distance entre deux enregistrements. Si aucune transformation n'était effectuée, les données initiales s'échelonnant sur de larges intervalles, comme par exemple les valeurs en dollars, prédomineraient toujours dans le calcul de la distance.

Description de la méthode employée

La procédure de transformation exécute les étapes suivantes pour chaque variable qui a été choisie au moins une fois comme champ d'appariement.

- Il classe toutes les valeurs acceptables de la variable par ordre croissant. Ces valeurs proviennent d'enregistrements donneurs, d'enregistrements receveurs dans lesquels la variable est un champ d'appariement, d'enregistrements receveurs dans lesquels la valeur de la variable est acceptable sans que cette dernière soit un champ d'appariement et d'enregistrements mixtes dans lesquels la valeur de la variable est acceptable. Le système ne tient pas compte des valeurs qui doivent être imputées, puisqu'elles ne sont pas acceptables.
- Le système attribue un rang à chaque variable. Dans le cas des groupes de valeurs égales, chaque valeur reçoit un rang, puis le système attribue à chaque enregistrement comportant cette valeur égale la moyenne de ces rangs.
- Le système divise alors chaque rang par le nombre total de valeurs acceptables plus un. Ce diviseur est susceptible d'être différent pour chaque variable, puisque le nombre d'enregistrements comportant des champs acceptables peut varier d'une variable à l'autre.

Exemple de transformation des valeurs

Valeurs initiales (classées)	Rang	Valeurs transformées
-12	1	.1
26	2	.2
38	4	.4
38	4	.4
38	4	.4
47	6.5	.65
47	6.5	.65
53	8	.8
105	9	.9

La transformation uniforme par rang permet d'obtenir des observations qui, à l'exception des valeurs égales, sont distribuées uniformément sur l'intervalle (0,1). Ainsi, les valeurs extrêmes sont rapprochées des autres observations et n'influent pas sur les valeurs transformées résultantes. On notera que, dans l'exemple ci-dessus, les valeurs transformées resteraient les mêmes si la valeur initiale la plus élevée était réduite à 54 ou augmentée à 1000.

Calcul de la distance

Lorsqu'on travaille avec des données numériques continues, on ne peut s'attendre à trouver une correspondance exacte entre un enregistrement receveur et un enregistrement donneur. Comme l'imputation par enregistrement donneur repose sur le transfert de données provenant d'un enregistrement qui ressemble le plus possible à l'enregistrement receveur, elle exige qu'on établisse une méthode visant à déterminer quel enregistrement est le plus proche, ou le plus similaire.

Banff utilise la norme L^∞ pour définir cette distance 4 entre deux enregistrements. La distance entre un enregistrement dont les champs d'appariement contiennent les valeurs transformées (x_1, x_2, \dots, x_n) et un enregistrement dont les champs d'appariement comportent les valeurs transformées (y_1, y_2, \dots, y_n) est définie par

$$\max(|x_1 - y_1|, |x_2 - y_2|, \dots, |x_n - y_n|).$$

Cette distance est souvent appelée distance minimax parce que l'enregistrement donneur le plus proche est celui pour lequel l'écart absolu maximum entre les valeurs transformées contenues dans ses champs d'appariement et celles contenues dans l'enregistrement receveur est le plus petit. Il y a lieu de noter que tous les champs d'appariement ont le même poids.

Exemple du calcul de la distance

Prenons par exemple les enregistrements suivants (un receveur et trois donneurs). On trouve ci-après les valeurs transformées contenues dans les trois champs d'appariement (x_1 , x_2 et x_3) de l'enregistrement receveur, ainsi que les distances entre l'enregistrement receveur et chaque enregistrement donneur.

	x_1	x_2	x_3	Distance jusqu'à l'enregistrement receveur
receveur	.5	.5	.5	
donneur 1	.6	.6	.6	$\max(.5-.6 , .5-.6 , .5-.6) = \max (.1, .1, .1) = .1$
donneur 2	.5	.5	.4	$\max(.5-.5 , .5-.5 , .5-.4) = \max (0, 0, .1) = .1$
donneur 3	.8	.5	.5	$\max(.5-.8 , .5-.5 , .5-.5) = \max (.3, 0, 0) = .3$

On notera que les enregistrements donneurs 1 et 2 sont à la même distance de l'enregistrement receveur, bien que la valeur contenue dans les trois champs de l'enregistrement donneur 1 soit à une distance de .1 de la valeur contenue dans les champs correspondants de l'enregistrement receveur, alors que la valeur comprise dans deux des champs de l'enregistrement donneur 2 est identique à celle qu'on trouve dans les champs correspondants de l'enregistrement receveur, seule la valeur de l'autre champ s'écartant de .1 de la valeur du champ correspondant. Les enregistrements donneurs 3 et 1 affichent le même écart absolu total par rapport à l'enregistrement receveur, mais cet écart est réparti entre les trois champs d'appariement dans le cas de l'enregistrement 1, tandis qu'il est concentré dans un seul champ d'appariement dans le cas de l'enregistrement 3. Selon cette définition de la distance, un enregistrement donneur dans lequel les valeurs contenues dans tous les champs d'appariement s'écartent de façon modérée de celles qui se trouvent dans les champs correspondants de l'enregistrement receveur est considéré comme plus proche de ce dernier qu'un enregistrement donneur pour lequel l'écart observé est très important dans le cas d'un seul champ d'appariement et minime ou nul dans le cas des autres.

7.4 EXÉCUTION DE L'IMPUTATION PAR ENREGISTREMENT DONNEUR

Objet

Cette fonction de Proc Donorimputation a pour objet de trouver, pour chaque enregistrement receveur, l'enregistrement donneur le plus proche qui contient des valeurs qui permettront à l'enregistrement receveur de satisfaire aux règles de vérification post-imputation spécifiées par l'utilisateur. Comme il est décrit dans Friedman, Bentley et Finkel (1977), une structure arborescente est créée afin de rendre la recherche des enregistrements donneurs plus efficace. Le système trouve des enregistrements donneurs tant pour les enregistrements comportant des champs d'appariement que pour ceux qui n'en comportent pas.

Description de la méthode employée

Un seul arbre est construit pour chaque groupe de données. La procédure exécute les étapes suivantes une fois pour l'ensemble des k champs d'appariement repérés précédemment par la fonction de détermination des champs d'appariement pour le groupe de données et le groupe de règles de vérification en cours de traitement.

- La procédure crée une structure arborescente pour organiser les enregistrements qu'il a désignés comme étant des enregistrements donneurs. Le nombre de niveaux que comporte l'arbre est fonction de la taille de compartiment. Vous trouverez ci-dessous une description plus détaillée de la manière dont un arbre est construit.
- Pour chaque enregistrement receveur comportant des champs d'appariement, la procédure parcourt l'arbre et choisit un groupe d'enregistrements donneurs les plus proches, dont la taille est le paramètre n spécifié par l'utilisateur.
- En commençant par le plus proche, la procédure met à l'essai les enregistrements donneurs faisant partie du groupe jusqu'à ce qu'il en trouve un dont les champs à imputer ne sont pas affectés d'un code FTE (valeur aberrante à exclure) et qui permettrait à l'enregistrement receveur de satisfaire aux règles de vérification post-imputation. S'il trouve un tel enregistrement, il procède à l'imputation et passe ensuite à l'enregistrement receveur suivant.
- Si aucun des n enregistrements donneurs possibles ne peut être utilisé, la procédure signale qu'aucun enregistrement donneur n'a pu être trouvé pour l'enregistrement receveur.
- Si un enregistrement receveur ne comporte pas de champs d'appariement, le système n'utilise pas l'arbre pour chercher un enregistrement donneur, mais l'utilisateur peut spécifier que la procédure choisit plutôt des enregistrements donneurs au hasard jusqu'à ce qu'il en trouve un qui convient, ce qui se produit normalement dès le premier enregistrement donneur choisi. On trouve une explication plus détaillée de cette opération ci-après.

Construction de l'arbre k-d

Il serait possible de chercher les enregistrements donneurs sans avoir recours à une structure arborescente, mais on accroît l'efficacité de la recherche en utilisant un arbre à k dimensions, où k est le nombre total de champs d'appariement dans le groupe de données et le groupe de règles de vérification. Lorsqu'il construit l'arbre, la procédure subdivise la population d'enregistrements donneurs en des groupes de plus en plus petits d'enregistrements semblables. De cette façon, les enregistrements donneurs qui sont proches d'un enregistrement receveur donné tendent à être

réunis dans certaines branches de l'arbre. De plus, il est ainsi possible d'exclure de la recherche certaines branches de l'arbre dont on peut déduire qu'elles ne peuvent contenir des enregistrements donneurs plus proches que ceux qui ont déjà été trouvés.

Chaque point où l'arbre se sépare en deux branches s'appelle un **noeud**. Le premier niveau, ou le plus élevé, appelé **noeud racine**, représente tous les enregistrements donneurs faisant partie de l'arbre. Au noeud racine, le système examine les k champs d'appariement utilisés dans le groupe de données et le groupe de règles de vérification et il choisit le champ pour lequel l'intervalle des valeurs transformées est le plus grand. Ce champ est appelé le **champ de branchement** du noeud racine. Le système choisit ensuite une **valeur de séparation** qui divisera les enregistrements donneurs en deux groupes les plus égaux possible. S'il existe un nombre pair d'enregistrements donneurs et si les valeurs centrales ne sont pas identiques, la valeur de séparation correspond à la médiane. Par contre, s'il existe un nombre impair d'enregistrements donneurs et si les valeurs centrales ne sont pas identiques, la valeur de séparation sera alors la moyenne des valeurs $\frac{n+1}{2}$ et $\frac{n+1}{2}-1$. Si les valeurs centrales sont identiques, Banff fait en sorte que les deux groupes soient les plus égaux possible tout en veillant à ce que les enregistrements aux valeurs identiques demeurent ensemble. En pareil cas, la valeur de séparation est la moyenne des valeurs centrales et de la valeur de l'enregistrement qui se trouvent immédiatement avant ou après le groupe d'enregistrements aux valeurs identiques. Une fois que le champ de branchement et les valeurs de séparation ont été calculées, le système forme au niveau suivant deux noeuds représentant chacun environ la moitié des enregistrements donneurs du niveau plus élevé.

Le choix des champs de branchement et des valeurs de séparation se poursuit jusqu'à ce que tous les noeuds représentent au maximum 16 enregistrements (la **taille du compartiment**). Les nœuds comportant un nombre d'enregistrements donneurs inférieur ou égal à 16 sont dits **noeuds terminaux**. Le noeud terminal ne peut représenter un nombre d'enregistrements donneurs supérieur à 16 si tous les champs d'appariement des enregistrements représentés par le noeud contiennent exactement les mêmes valeurs. Dans ce cas, le système attribuerait les enregistrements donneurs aux valeurs identiques au même noeud terminal, même si leur nombre est supérieur à 16. L'ensemble des noeuds terminaux permet de subdiviser l'espace à k dimensions en des sous-ensembles mutuellement exclusifs et exhaustifs d'enregistrements donneurs.

Exemple de construction d'un arbre

Prenons par exemple huit enregistrements donneurs qui doivent être placés dans un arbre k-d pour lequel l'utilisateur a spécifié une taille de compartiment égale à deux. Le choix d'une si petite taille de compartiment s'explique par la nécessité de garder le présent exemple maniable. En pratique, Banff utilise toujours une taille de 16. On notera que la liste qui suit ne contient que des enregistrements donneurs; il se peut donc que des valeurs transformées appartenant à des enregistrements receveurs se situent entre celles qui apparaissent ci-après.

	Valeurs transformées	
	x	y
enregistrement donneur 1	.10	.20
enregistrement donneur 2	.30	.50
enregistrement donneur 3	.40	.60
enregistrement donneur 4	.50	.55
enregistrement donneur 5	.50	.80
enregistrement donneur 6	.50	.70
enregistrement donneur 7	.85	.95
enregistrement donneur 8	.95	.60

Les lettres utilisées pour désigner chaque noeud dans la description qui suit sont les mêmes que celles qui sont employées dans la figure 7.3. À chaque noeud, la première étape de la construction de l'arbre consiste à décider quelle variable doit servir de champ de branchement. Comme les variables x et y constituent les seuls champs d'appariement dans le présent exemple, le champ de branchement sera toujours un de ces deux champs. Au noeud racine, le noeud A, l'intervalle des valeurs prises par x dans tous les enregistrements donneurs a une amplitude de $.95 - .10 = .85$, et l'intervalle des valeurs prises par y dans les mêmes enregistrements a une amplitude de $.95 - .20 = .75$. Le système choisit donc x comme champ de branchement. L'étape suivante consiste à séparer les enregistrements donneurs en deux groupes les plus uniformes possible. Normalement, la séparation se ferait entre les enregistrements donneurs 4 et 5. Toutefois, comme les enregistrements 4, 5 et 6 comportent les mêmes valeurs pour x, la séparation doit se faire juste avant ou juste après ce groupe, c'est-à-dire soit entre les enregistrements donneurs 3 et 4, soit entre les enregistrements donneurs 6 et 7. La séparation la plus égale est celle qui est faite entre les enregistrements 3 et 4. La valeur de séparation calculée est .45, soit la moyenne des valeurs de x dans les enregistrements donneurs 3 et 4. Tous les enregistrements dans lesquels la valeur de x est inférieure ou égale à .45 sont rattachés au noeud B, tandis que ceux dans lesquels la valeur de x est supérieure à .45 sont rattachés au noeud C. Le noeud B représente tous les enregistrements donneurs dans lesquels la valeur de x est inférieure ou égale à .45, alors que le noeud C représente tous les enregistrements donneurs dans lesquels la valeur de x est supérieure à .45. L'égalité est toujours attribuée au noeud de gauche. Il est possible que x prenne une valeur égale à .45 dans un enregistrement receveur, mais on ne peut pas trouver cette valeur dans un enregistrement donneur, puisque la valeur de séparation est la moyenne des valeurs contenues dans deux enregistrements donneurs consécutifs.

Le noeud B représente maintenant trois enregistrements (les enregistrements donneurs 1, 2 et 3): il ne peut donc pas être un noeud terminal puisque l'utilisateur a spécifié que le compartiment terminal devait avoir une taille égale à deux. L'amplitude de l'intervalle des valeurs prises par x dans ces enregistrements est de .30 et celle de l'intervalle des valeurs prises par y est de .40: le système choisit donc y comme champ de branchement pour ce noeud. Il est impossible de diviser un nombre impair d'enregistrements exactement en deux. En pareil cas, Banff effectue toujours la séparation entre la valeur centrale et la prochaine valeur plus élevée. Dans le présent exemple, il y a trois enregistrements et Banff effectue la séparation entre le deuxième et le troisième. Les enregistrements donneurs 1 et 2 sont rattachés au noeud D et l'enregistrement donneur 3 au noeud E. Comme ces deux noeuds ne représentent chacun pas plus de deux enregistrements, ils deviennent tous deux des noeuds terminaux. Les variables x et y prennent des valeurs inférieures ou égales à .45 et à .55 respectivement dans tous les enregistrements rattachés au noeud D, tandis que x prend une valeur inférieure ou égale à .45 et y, une valeur supérieure à .55, dans tous les enregistrements rattachés au noeud E.

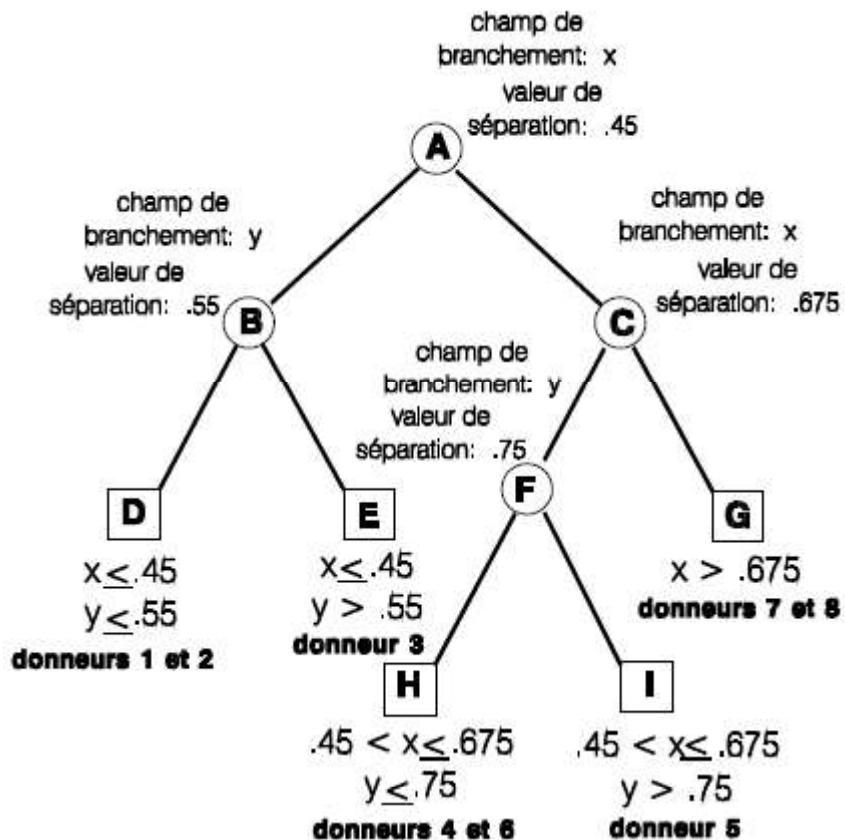


Figure 7.3 Exemple de la construction d'un arbre k-d

La construction de l'arbre se poursuit au noeud C. L'amplitude de l'intervalle des valeurs de la variable x étant de .45 et celle de l'intervalle de y étant de .40, Banff choisit x comme champ de branchement. Il importe peu que x ait aussi été choisi comme champ de branchement au noeud A. Le système établit la valeur de séparation des enregistrements donneurs à .675; il attribue donc les enregistrements donneurs 4, 5 et 6 au noeud F et les enregistrements donneurs 7 et 8 au noeud G. Ce dernier devient un noeud terminal parce qu'il ne représente que deux enregistrements. La variable x doit prendre une valeur supérieure à .675 dans tous les enregistrements rattachés au noeud G, mais la variable y peut y prendre n'importe quelle valeur située entre 0 et 1.

Au noeud F, le système se sert des valeurs de y pour séparer les enregistrements donneurs 4, 5 et 6 parce que c'est leur intervalle qui possède la plus grande amplitude. La valeur de séparation est la moyenne des valeurs de y dans les enregistrements donneurs 5 et 6. Le système attribue les enregistrements donneurs 4 et 6 au noeud terminal H et l'enregistrement donneur 5 au noeud terminal I. Dans les enregistrements rattachés au noeud terminal H, la valeur de x doit être supérieure à .45 et inférieure ou égale à .675 et la valeur de y, inférieure ou égale à .75. Dans les enregistrements rattachés au noeud terminal I, la valeur de x doit être supérieure à .45 et inférieure ou égale à .675 et la valeur de y, supérieure à .75.

Enregistrements receveurs comportant des champs d'appariement - Parcours de l'arbre

Supposons maintenant que Banff parcourt l'arbre qui vient d'être construit dans l'exemple précédent pour trouver l'enregistrement donneur le plus proche pour un enregistrement receveur dans lequel les valeurs transformées de x et y sont .53 et .74 respectivement. Une valeur de deux sera utilisée pour le paramètre n . Cette valeur est trop peu élevée pour être employée dans la plupart des applications, mais on l'a choisie dans le présent cas afin que l'exemple soit d'une taille maniable. Comme il n'y a que deux champs d'appariement dans cet exemple, le partitionnement de la population d'enregistrements donneurs peut être représenté par un graphique à 2 dimensions. La figure 7.4 illustre les régions du plan qui sont représentées par chacun des noeuds terminaux. Les enregistrements donneurs sont désignés par un « x » et l'enregistrement receveur par un « $*$ ». Les figures 7.3 et 7.4 illustrent différentes manières de représenter le même arbre k-d. La figure 7.3 fait ressortir le rapport hiérarchique entre les noeuds, tandis que la figure 7.4 met l'accent sur la façon dont les noeuds terminaux partitionnent l'espace défini par les enregistrements donneurs. La figure 7.3 est la manière habituelle de représenter l'arbre k-d; les graphiques comme la figure 7.4 ne peuvent pas être représentés dans les dimensions plus élevées qui seraient nécessaires pour pratiquement toutes les applications.

Au début de la recherche, l'ensemble de n enregistrements donneurs les plus proches est vide. Toutefois, l'ensemble va acquérir des éléments et, à mesure que la recherche avance, il se peut que certains des éléments initiaux soient remplacés par des enregistrements donneurs qui sont encore plus proches de l'enregistrement receveur.

La recherche commence au noeud A, le noeud racine. Le noeud A n'étant pas un noeud terminal, le système passe au noeud C, puisque la valeur de x dans l'enregistrement receveur est supérieure à .45 et se situe donc dans l'intervalle représenté par le noeud C. Le noeud C n'est pas un noeud terminal; le système poursuit donc la recherche. La valeur de x , le champ de branchement, dans l'enregistrement receveur étant inférieure à .675, le système passe au noeud F. Ce noeud n'étant pas un noeud terminal, la recherche se poursuit jusqu'au noeud H, parce que la valeur de y dans l'enregistrement receveur est inférieure à .75. Comme ce noeud est le premier noeud terminal que le système rencontre depuis le début de la recherche, l'ensemble de n enregistrements donneurs les plus proches est donc toujours vide. Étant donné que le paramètre n est égal à deux et qu'il y a deux enregistrements donneurs représentés par le noeud H, le système entre ces deux enregistrements dans l'ensemble des enregistrements donneurs les plus proches.

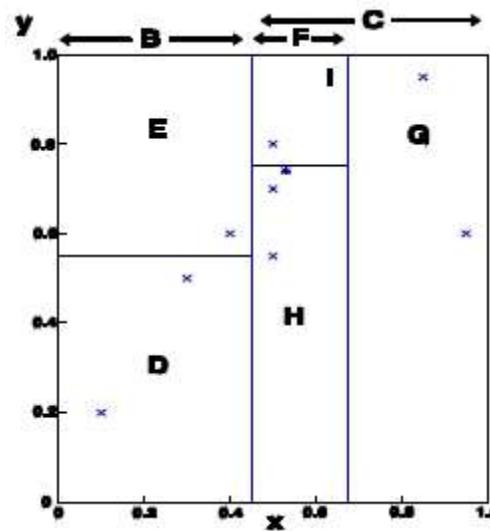


Figure 7.4 Les noeuds terminaux partitionnent l'espace défini par les enregistrements donneurs

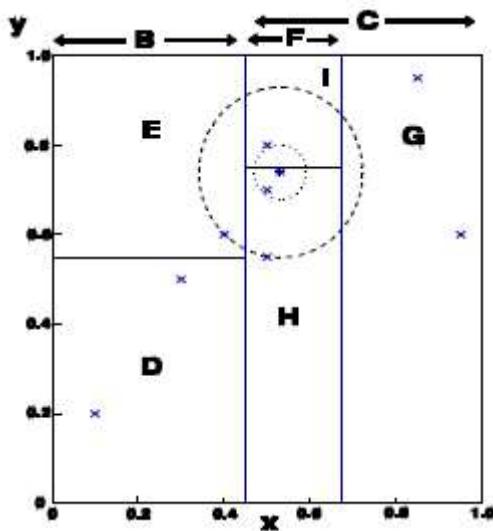


Figure 7.5 Utilisation du test
« bounds-overlap-ball »

enregistrements les plus proches, l'enregistrement donneur qui est le plus éloigné de l'enregistrement receveur est l'enregistrement 4, qui s'en trouve à une distance de .19. La ligne discontinue représentant un cercle d'un rayon de .19 dont le centre est l'enregistrement receveur (.53, .74) chevauche les limites du noeud I, comme on peut le voir à la figure 7.5. De fait, il se peut qu'il y ait au noeud I des enregistrements donneurs qui ne s'écartent de l'enregistrement receveur que d'une distance légèrement supérieure à .01. Ce pourrait être le cas si, par exemple, x prenait, dans un enregistrement donneur où la valeur de y est un peu plus élevée que .75, la même valeur que dans l'enregistrement receveur. Le système doit donc explorer le noeud I puisqu'il est possible qu'il y trouve un enregistrement donneur qui est plus proche de l'enregistrement receveur que les enregistrements donneurs qui font présentement partie de l'ensemble de n enregistrements donneurs les plus proches.

Comme le noeud I est un noeud terminal, le système compare la distance entre son seul enregistrement donneur (l'enregistrement 5) et l'enregistrement receveur à celle entre ce dernier et les enregistrements donneurs qui se trouvent déjà dans l'ensemble de n enregistrements donneurs les plus proches. Cette distance (.06) étant inférieure à celle entre l'enregistrement receveur et l'enregistrement donneur 4, le système remplace ce dernier par l'enregistrement donneur 5 dans l'ensemble.

Le système poursuit sa recherche en revenant au noeud F et, comme il a déjà examiné les noeuds H et I, il doit maintenant décider s'il doit ou non explorer le noeud G. À cette fin, il exécute le test « boundsoverlap-ball », comme il l'a fait au noeud H. L'enregistrement donneur 5, qui se trouve à une distance de .06, est maintenant celui qui est le plus éloigné dans le groupe de n enregistrements donneurs les plus proches. Dans la figure 7.5, on peut voir que la ligne pointillée représentant un cercle d'un rayon de .06 dont le centre est l'enregistrement receveur ne chevauche pas les limites du noeud G. Comme la variable x prend une valeur supérieure à .675 dans tous les enregistrements donneurs rattachés au noeud G, tous ces enregistrements doivent donc se trouver à une distance de .145 (.675 - .530) de l'enregistrement receveur, même si la variable y prend exactement les mêmes valeurs que dans l'enregistrement receveur. Par conséquent, il serait inutile d'explorer le noeud G.

L'étape suivante consiste à décider s'il peut y avoir ou non des enregistrements donneurs appartenant au noeud opposé, le noeud I, qui sont plus proches de l'enregistrement receveur que ceux qui ont déjà été choisis. Cette décision repose sur les résultats du test « bounds-overlap-ball » qu'ont décrit Friedman, Bentley et Finkel (1977). Ce test permet de faire en sorte qu'un noeud ne fasse pas l'objet d'une recherche si l'enregistrement donneur le plus proche possible représenté par ce noeud est plus éloigné de l'enregistrement receveur que l'enregistrement donneur le plus éloigné qui a déjà été choisi dans le groupe de n enregistrements donneurs les plus proches. Grâce à ce test, le système peut déterminer si les limites du noeud opposé chevauchent un cercle dont le centre est l'enregistrement receveur et dont le rayon est la distance entre ce dernier et l'enregistrement donneur le plus éloigné dans le groupe courant de n enregistrements. Dans l'ensemble courant des

Le système revient au noeud C et, comme il a déjà examiné le noeud F et soumis le noeud G au test, il doit maintenant décider s'il doit ou non explorer le noeud B. L'enregistrement donneur 5, qui se trouve à une distance de .06, demeure celui qui est le plus éloigné dans le groupe de n enregistrements donneurs les plus proches. Dans la figure 7.5, on peut voir que la ligne pointillée représentant un cercle d'un rayon de .06 dont le centre est l'enregistrement receveur ne chevauche pas les limites du noeud B; il n'est donc pas nécessaire d'explorer cette branche de l'arbre. Comme la variable x doit prendre une valeur inférieure ou égale à .45 dans les enregistrements donneurs appartenant au noeud B, ceux-ci doivent se situer à une distance d'au moins .08 de l'enregistrement receveur. C'est ainsi que se termine l'exploration de l'arbre. Les enregistrements donneurs 5 et 6 sont les membres de l'ensemble de n enregistrements donneurs les plus proches.

Une fois le groupe de n enregistrements donneurs les plus proches réuni, la procédure examine l'enregistrement donneur le plus proche pour vérifier s'il ne comporte pas de champs à imputer auxquels des codes FTE (valeur à exclure) ont été attribués et pour déterminer si l'enregistrement receveur satisferait aux règles de vérification post-imputation si on transférait dans les champs à imputer les valeurs contenues dans l'enregistrement donneur le plus proche. Si ces vérifications donnent les résultats escomptés, le système procède à l'imputation, puis il traite l'enregistrement receveur suivant. Si les valeurs contenues dans l'enregistrement donneur le plus proche ne permettent pas à l'enregistrement receveur de satisfaire aux règles de vérification post-imputation, le système essaie avec le prochain enregistrement donneur le plus proche. Il poursuit ce processus jusqu'à ce qu'il trouve un enregistrement donneur ou jusqu'à ce que l'ensemble des n enregistrements donneurs faisant partie du groupe se soient avérés non convenables.

Dans le petit exemple ci-dessus, le champ de branchement faisait toujours partie du groupe de champs d'appariement de l'enregistrement receveur. En pratique, il y aurait probablement de nombreuses situations où ce ne serait pas le cas. Dans ces situations, le système doit explorer les parties de l'arbre se trouvant à droite et à gauche du noeud en question parce que le champ de branchement ne limite pas les enregistrements donneurs qui conviennent pour un enregistrement receveur ne comportant pas ce champ d'appariement. Dans ce cas, Banff effectue toujours la recherche en descendant du côté gauche de l'arbre, puis en remontant du côté droit.

Enregistrements receveurs ne comportant aucun champ d'appariement

La procédure ne se sert pas de l'arbre k-d pour chercher des enregistrements donneurs lorsque l'enregistrement receveur qu'il traite ne comporte pas de champs d'appariement. À la place, l'utilisateur peut spécifier que le système choisit un enregistrement au hasard dans la population d'enregistrements donneurs. Si cette option n'est pas choisie, Banff ne cherche pas un donneur. Si l'option est choisie, Banff sélectionne le donneur au hasard et puis il soumet ensuite l'enregistrement à une vérification pour déterminer s'il y a au moins un champ avec une valeur à exclure (FTE) qui correspond à l'un des champs à imputer de l'enregistrement receveur. Si c'est le cas, le système ne peut pas utiliser cet enregistrement donneur et il en tire un autre au hasard de l'ensemble réduit d'enregistrements donneurs. Sinon, Banff procède à l'imputation si l'enregistrement qui en résulte satisfait aux règles de vérification post-imputation.

Malgré les apparences, il est peu probable que la recherche aléatoire d'un enregistrement donneur convenable s'étende en longueur, en raison d'une caractéristique des enregistrements receveurs ne comportant pas de champs d'appariement. On se rappellera que lorsque les valeurs acceptables contenues dans l'enregistrement receveur sont substituées aux variables des règles de vérification initiales, les règles qui définissent l'ensemble réduit ne comprennent aucune variable autre que celles qui doivent faire l'objet d'une imputation. Comme tous les enregistrements donneurs ont satisfait aux règles de vérification initiales, les champs devant être transférés doivent être

cohérents entre eux et le demeureront après leur transfert en groupe dans l'enregistrement receveur. De plus, ces champs ne peuvent pas provoquer le rejet de l'enregistrement receveur en vertu d'autres règles de vérification puisqu'aucun des champs acceptables n'est visé par celles-ci, compte tenu des valeurs réelles contenues dans cet enregistrement receveur.

Cette relation n'est vraie que si la région d'acceptation définie par les règles de vérification initiales est entièrement contenue dans celle qui est définie par les règles post-imputation. Si certaines parties de la région d'acceptation initiale se trouvent à l'extérieur de la région post-imputation, un enregistrement donneur qui se situe dans l'une de ces parties ne peut pas être utilisé avec succès pour imputer certains enregistrements receveurs puisque la condition imposée par les règles post-imputation à l'enregistrement receveur est plus restrictive que celle qui était imposée à l'enregistrement donneur lorsque ce dernier a satisfait aux règles initiales.

Règles de vérification post-imputation

Banff n'impose aucune contrainte sur les règles de vérification post-imputation, si ce n'est que toutes les variables visées par les règles initiales doivent aussi être visées par les règles post-imputation, et vice versa, même dans le cas où les règles limitent les variables en-dessus ou en-dessous. Les règles postimputation sont habituellement moins strictes que les règles initiales, bien que l'utilisateur puisse choisir de les faire autant, sinon plus restrictives. Il n'est pas rare de remplacer une égalité dans les règles de vérification initiales par deux inégalités qui définissent les limites supérieure et inférieure à l'intérieur desquelles le total doit se situer. Autrement, dans le cas des enregistrements qui n'ont pas satisfait à la règle initiale fondée sur une égalité, la procédure d'imputation par enregistrement donneur poursuivra sa recherche jusqu'à ce qu'il trouve un enregistrement donneur qui comporte une valeur qui satisfasse exactement à cette règle. Il pourrait ainsi rejeter de nombreux enregistrements donneurs éventuels qui sont proches de l'enregistrement receveur et en choisir un qui en est éloigné, mais qui se trouve à contenir la seule valeur requise. Il se pourrait également qu'il ne trouve aucun enregistrement donneur comportant la « bonne » valeur.

8. PROC ESTIMATOR – IMPUTATION PAR ESTIMATEUR

Objet

La procédure permet d'imputer une variable à la fois en utilisant toute une variété d'estimateurs. L'utilisateur peut choisir parmi les 20 algorithmes pré-définis dans le système ou définir ses propres algorithmes personnalisés (les termes « estimateur » et « algorithme » sont interchangeables). Deux types d'algorithmes sont disponibles dans Banff: fonctions d'estimation et estimateurs par régression linéaire. Ces algorithmes peuvent utiliser des données courantes et/ou historiques. La procédure considère toutes les données historiques comme étant correctes. L'utilisateur peut choisir de calculer les paramètres requis (moyennes et coefficients de régression) à partir de l'ensemble des valeurs acceptables que prend la variable ou définir un sous-ensemble d'enregistrements à utiliser pour le calcul des paramètres. Comme les algorithmes sont appliqués de façon indépendante aux variables choisies, il est possible que les enregistrements imputés résultants ne satisfassent pas les règles de vérification initiales. L'utilisateur peut soumettre ces enregistrements imputés à la procédure de localisation des erreurs ou leur propre processus de vérification afin de déterminer lesquels d'entre eux satisfont les règles de vérification.

Types d'algorithmes

Les **fonctions d'estimation** sont des expressions mathématiques impliquant des valeurs courantes ou historiques des variables de l'enregistrement à imputer, ainsi que des moyennes courantes ou historiques. Elles peuvent également comprendre des parenthèses et les opérateurs arithmétiques habituels: addition (+), soustraction (-), multiplication (*), division (/) et exponentiation (^).

Les **estimateurs par régression linéaire** déterminent la valeur de l'imputation à l'aide de modèles de régression linéaire dont la forme générale est

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1 T_1}^{p_1} + \hat{\beta}_2 x_{i2 T_2}^{p_2} + \hat{\beta}_3 x_{i3 T_3}^{p_3} + \cdots + \hat{\beta}_m x_{im T_m}^{p_m} + \hat{\epsilon}_i$$

où les T_j font référence à des périodes courantes ou historiques et les p_j sont des exposants. La variable à imputer y_i est la variable dépendante du modèle alors que les variables auxiliaires x_{ij} sont les variables indépendantes, ou régresseurs. Les $\hat{\beta}_j$ sont les coefficients de régression et ils sont résolus par le système à l'aide de la méthode des moindres carrés. Le $\hat{\epsilon}_i$ est un terme d'erreur aléatoire que l'utilisateur peut ajouter au modèle afin d'introduire une certaine variabilité dans les données imputées.

Description de la méthode employée

Chaque fois qu'il choisit d'appliquer la méthode d'imputation par estimateur à une variable, l'utilisateur doit spécifier plusieurs options. La procédure peut traiter plusieurs variables à la fois pendant une exécution. L'utilisateur peut spécifier plus d'un algorithme pour une variable pendant une exécution de la procédure. Si le premier algorithme défini ne peut pas calculer une valeur pour une variable, Banff va essayer le deuxième algorithme et ainsi de suite.

Les paramètres requis sont calculés avant l'exécution de la procédure. Les valeurs de ces paramètres ne changeront pas pendant l'exécution de la procédure. C'est-à-dire que les paramètres ne sont pas recalculés pendant l'exécution pour tenir compte des nouvelles valeurs imputées. Le calcul des paramètres est fait avant chaque nouvelle exécution de la procédure, et

toutes les valeurs imputées précédemment sont disponibles pour le calcul.

Définition de l'estimateur

Voici la liste des renseignements devant être introduits en fichier SAS pour chaque exécution de Proc Estimator.

- Algorithme, identificateur de champ et variable auxiliaire - Ces renseignements servent à décrire l'imputation de base devant être exécutée. L'utilisateur précise l'algorithme à être utilisé, la variable devant faire l'objet de l'imputation et les variables auxiliaires, si l'algorithme choisi en fait usage.
- Variable de pondération - Si l'algorithme choisi utilise des paramètres, l'utilisateur peut désigner une variable devant servir de coefficient de pondération pour les fins du calcul de ces paramètres. L'utilisation d'une variable de pondération est étudiée de façon plus approfondie dans la suite de la présente section.
- Variable de variance, exposant de variance, période de variance - l'utilisateur peut inclure une variable de variance du modèle pour la spécification d'un estimateur par régression linéaire. Cette variable est prise en considération lors du calcul des coefficients de régression et de l'erreur aléatoire. L'utilisation d'une variable de variance est discutée en plus de détail dans la suite de la présente section.
- Terme d'erreur aléatoire - L'utilisateur peut inclure un terme d'erreur aléatoire dans la spécification de l'estimateur. Le terme d'erreur est ajouté à la valeur imputée afin de créer dans les données imputées la même variabilité que dans les données non imputées.
- Exclusion du calcul des paramètres - Si l'algorithme choisi utilise des paramètres, l'utilisateur a la possibilité de déterminer dans une certaine mesure quels enregistrements serviront au calcul de ces paramètres. Il lui est possible d'exclure de ce calcul les champs qui comportent des valeurs à exclure (FTE), les champs qui ont déjà fait l'objet d'une imputation ainsi que les autres données spécifiées par l'utilisateur. Ces options sont étudiées de façon plus détaillée ci-après.
- Critères relatifs au calcul des paramètres - Si l'algorithme choisi utilise des paramètres, l'utilisateur doit spécifier le pourcentage et le nombre d'enregistrements minimaux dont doit disposer le système pour calculer les paramètres. Ces critères sont étudiés de façon plus détaillée ci-après.

Description des algorithmes pré-définis dans Banff

Soit y_{iC} la variable à imputer pour l'unité i au temps C (période courante);

y_{iH} la variable à imputer pour l'unité i au temps H (période historique);

x_{iC}, x_{iH} est la variable auxiliaire pour l'unité i au temps C ou H, respectivement.

$u_{it}, v_{it}, w_{it}, z_{it}$	d'autres variables auxiliaires pour l'unité i au temps T (période courante ou historique);
\bar{y}_C, \bar{y}_H	les moyennes des valeurs prises par la variable à imputer dans tous les enregistrements admissibles des fichiers courant et historique respectivement;
\bar{x}_C, \bar{x}_H	les moyennes des valeurs prises par la variable auxiliaire dans tous les enregistrements admissibles des fichiers courant et historique respectivement;
$\hat{\beta}_j$	le j ^e coefficient de régression du modèle calculé en utilisant tous les enregistrements admissibles des fichiers courant et/ou historique.

Le calcul des moyennes et des coefficients de régression est basé sur tous les enregistrements admissibles, c.-à-d., les enregistrements restants après que les exclusions aient été appliquées et que Banff se soit assuré que le calcul des moyennes et des coefficients de régression est basé sur les mêmes enregistrements. L'imputation n'est pas effectuée si le nombre d'enregistrements admissibles ne satisfait pas les critères relatifs au calcul des paramètres ou si les valeurs, courantes ou historiques, des variables auxiliaires requises par l'algorithme ne sont pas valides (manquantes, négatives ou nécessitant une imputation) pour l'enregistrement à imputer.

Les 20 algorithmes suivants sont pré-définis dans Banff. Le format définit le comportement de l'algorithme. L'utilisateur pourra se référer au format des algorithmes pré-définis quand viendra le temps de définir ses propres algorithmes.

Fonctions d'estimation

Algorithme: AUXTREND

Équation:

$$\hat{y}_{iC} = \frac{x_{iC}}{x_{iH}} y_{iH}$$

Format: aux1(c,v) * fieldid(h,v) / aux1(h,v)

Description: La valeur de la même unité lors du cycle d'enquête précédent, corrigée en fonction de la variation d'une variable auxiliaire, est imputée

Algorithme: AUXTREND2

Équation:

$$\hat{y}_{iC} = \frac{y_{iH}}{2} \left(\frac{u_{iC}}{u_{iH}} + \frac{v_{iC}}{v_{iH}} \right)$$

Format: fieldid(h,v) / 2 * (aux1(c,v)/aux1(h,v) + aux2(c,v)/aux2(h,v))

Description: Une moyenne de deux AUXTRENDs est imputée

Algorithme: CURAUX

Équation: $\hat{y}_{iC} = x_{iC}$

Format: aux1(c,v)

Description: La valeur courante d'une variable auxiliaire pour la même unité est imputée

Algorithme:	CURAUXMEAN
Équation:	$\hat{y}_{iC} = \bar{x}_C$
Format:	aux1(c,a)
Description:	La moyenne courante d'une variable auxiliaire est imputée
Algorithme:	CURMEAN
Équation:	$\hat{y}_{iC} = \bar{y}_C$
Format:	fieldid(c,a)
Description:	La valeur moyenne de la variable à imputer pour toutes les unités admissibles à l'occasion du cycle d'enquête courant est imputée
Algorithme:	CURRATIO
Équation:	$\hat{y}_{iC} = \frac{\bar{y}_C}{\bar{x}_C} x_{iC}$
Format:	fieldid(c,a) * aux1(c,v) / aux1(c,a)
Description:	Une estimation par quotient, établie à partir des moyennes calculées pour les unités admissibles à l'occasion du cycle d'enquête courant, est imputée.
Algorithme:	CURRATIO2
Équation:	$\hat{y}_{iC} = \frac{\bar{y}_C}{2} \left(\frac{u_{iC}}{\bar{u}_C} + \frac{v_{iC}}{\bar{v}_C} \right)$
Format:	fieldid(c,a)/2 * (aux1(c,v)/aux1(c,a) + aux2(c,v)/aux2(c,a))
Description:	Une moyenne de deux CURRATIOs est imputée
Algorithme:	CURSUM2
Équation:	$\hat{y}_{iC} = u_{iC} + v_{iC}$
Format:	aux1 + aux2
Description:	La somme de deux variables auxiliaires provenant du cycle d'enquête courant est imputée
Algorithme:	CURSUM3
Équation:	$\hat{y}_{iC} = u_{iC} + v_{iC} + w_{iC}$
Format:	aux1 + aux2 + aux3
Description:	La somme de trois variables auxiliaires provenant du cycle d'enquête courant est imputée
Algorithme:	CURSUM4
Équation:	$\hat{y}_{iC} = u_{iC} + v_{iC} + w_{iC} + z_{iC}$
Format:	aux1 + aux2 + aux3 + aux4
Description:	La somme de quatre variables auxiliaires provenant du cycle d'enquête courant est imputée

Algorithme:	DIFTREND
Équation:	$\hat{y}_{iC} = \frac{\bar{y}_C}{\bar{y}_H} y_{iH}$
Format:	fieldid(c,a) * fieldid(h,v) / fieldid(h,a)
Description:	La valeur de la même unité lors du cycle d'enquête précédent, corrigée en fonction de la variation de la moyenne de cette variable, est imputée
Algorithme:	PREAUX
Équation:	$\hat{y}_{iC} = x_{iH}$
Format:	aux1(h,v)
Description:	La valeur historique d'une variable auxiliaire pour la même unité est imputée
Algorithme:	PREAUXMEAN
Équation:	$\hat{y}_{iC} = \bar{x}_H$
Format:	aux1(h,a)
Description:	La moyenne historique d'une variable auxiliaire est imputée
Algorithme:	PREMEEAN
Équation:	$\hat{y}_{iC} = \bar{y}_H$
Format:	fieldid(h,a)
Description:	La valeur moyenne de la variable à imputer pour toutes les unités admissibles à l'occasion du cycle d'enquête précédent est imputée
Algorithme:	PREVALUE
Équation:	$\hat{y}_{iC} = y_{iH}$
Format:	fieldid(h,v)
Description:	La valeur de la même unité lors du cycle d'enquête précédent est imputée

Estimateurs par régression linéaire

Algorithme:	CURREG
Équation:	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 x_{iC}$
Format:	intercept, aux1(c)
Description:	Régression linéaire simple basée sur une variable indépendante provenant du cycle d'enquête courant
Algorithme:	CURREG_E2
Équation:	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 x_{iC} + \hat{\beta}_2 x_{iC}^2$
Format:	intercept, aux1(c), aux1(c) ²
Description:	Régression linéaire basée sur la valeur et le carré de la valeur d'une variable provenant du cycle d'enquête courant

Algorithme:	CURREG2
Équation:	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 u_{iC} + \hat{\beta}_2 v_{iC}$
Format:	intercept, aux1(c), aux2(c)
Description:	Régression linéaire basée sur deux variables indépendantes provenant du cycle d'enquête courant
Algorithme:	CURREG3
Équation:	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 u_{iC} + \hat{\beta}_2 v_{iC} + \hat{\beta}_3 w_{iC}$
Format:	intercept, aux1(c), aux2(c), aux3(c)
Description:	Régression linéaire basée sur trois variables indépendantes provenant du cycle d'enquête courant
Algorithme:	HISTREG
Équation:	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 y_{iH}$
Format:	intercept, fieldid(h)
Description:	Régression linéaire basée sur les valeurs prises par la variable à imputer lors du cycle d'enquête précédent

Algorithmes définis par l'utilisateur

L'utilisateur peut choisir de définir un ou plusieurs algorithmes personnalisés pour l'imputation par estimateur. Ainsi, en plus d'avoir accès aux 20 algorithmes pré-définis dans Banff, l'utilisateur pourra également choisir parmi ses algorithmes personnalisés. La définition de ces algorithmes doit être faite avant la spécification au système des algorithmes, pré-définis ou personnalisés, à être utilisés pour l'imputation.

Plusieurs renseignements doivent être spécifiés lors de la définition d'algorithmes personnalisés. L'utilisateur doit conserver ces informations dans un fichier SAS.

- Nom: l'utilisateur peut identifier uniquement l'algorithme qu'il est en train de définir en spécifiant dans ce champ un nom composé de caractères alphanumériques.
- Type: le type de l'algorithme est identifié dans ce champ. Les valeurs admissibles sont « EF » pour une fonction d'estimation (« Estimator Function ») et « LR » pour une régression linéaire (« Linear Regression »).
- Statut: l'utilisateur spécifie dans ce champ un code alphanumérique qui sera associé à l'algorithme. Ce code est écrit au fichier de sortie de statut des champs en notant la méthode d'imputation qui a été utilisée pour imputer un champ. Notez que Banff ajoute automatiquement un 'I', pour imputation, devant le code.
- Formule : le comportement de l'algorithme est défini dans ce champ en utilisant une combinaison de noms de variables, de mots réservés, et d'opérateurs mathématiques. La syntaxe dépend du type de l'algorithme.
- Description : des commentaires au sujet de l'algorithme peuvent être ajoutés dans ce Champ.

Moyennes pondérées et non pondérées

L'utilisateur peut spécifier une variable qui sera utilisée comme coefficient de pondération pour les fins du calcul des paramètres d'un ou plusieurs estimateurs. Il peut être approprié de spécifier une telle variable si, par exemple, les enregistrements faisant partie du groupe de données soumis à l'imputation avaient des probabilités de sélection inégales dans le plan d'échantillonnage initial. Lorsque l'utilisateur spécifie une variable de pondération, toutes les moyennes utilisées par l'algorithme en question sont des moyennes pondérées calculées comme suit :

$$\bar{y}_C = \frac{\sum_i w_{iC} y_{iC}}{\sum_i w_{iC}}, \quad \bar{y}_H = \frac{\sum_i w_{iH} y_{iH}}{\sum_i w_{iH}}$$

Similairement, cette variable de pondération est prise en considération dans le calcul de coefficients de régression.

Le poids spécifié peut être n'importe laquelle des variables figurant dans le fichier SAS de données courantes. Il incombe à l'utilisateur de s'assurer que la valeur déclarée dans le champ de pondération est valide. Si la variable de pondération contient des valeurs négatives ou manquantes, Banff va interrompre l'exécution de la procédure, et ce, avant d'avoir tenté toute imputation. Par ailleurs, si l'algorithme nécessite le calcul de la moyenne des valeurs historiques et celui de la moyenne des valeurs courantes, le système utilise automatiquement la même variable comme coefficient de pondération pour les deux cycles d'enquête. Le système calcule la moyenne des valeurs historiques à partir des valeurs de la variable de pondération tirées de la variable correspondante du fichier de données historiques.

Variable de variance du modèle dans les régressions linéaires

Lors de la définition d'un estimateur par régression linéaire, l'utilisateur peut spécifier une variable de variance du modèle. Cette variable est prise en considération lors du calcul des coefficients de régression et de l'erreur aléatoire. La variance de la variable imputée est proportionnelle à la variable de variance du modèle. Autrement dit, la variance de la variable imputée est estimée par $v_i \sigma^2$, où v_i est la valeur de la variable de variance du modèle pour l'enregistrement i et σ^2 est la variance de la population. La variance σ^2 est présumée égale pour tous les enregistrements du groupe de données mais n'a pas à être connue.

Il revient à l'utilisateur de s'assurer que la variable de variance du modèle déclarée existe et contienne des valeurs valides. Si la variable de variance du modèle contient des valeurs nulles, négatives ou manquantes, Banff va interrompre l'exécution de la procédure, et ce, avant d'avoir tenté toute imputation.

Terme d'erreur aléatoire dans les régressions linéaires

Lors de la définition d'un estimateur, que ce soit une fonction d'estimation ou une régression linéaire, l'utilisateur peut ajouter une erreur aléatoire au modèle. Cette erreur aléatoire, ou résidu, est ajoutée à la valeur de la variable imputée obtenue du modèle de régression afin de créer dans les données imputées la même variabilité que dans les données non imputées.

Si une erreur aléatoire est spécifiée, Banff calcule d'abord la valeur de la variable à être imputée en utilisant l'estimateur par régression choisi. Ensuite, le système sélectionne aléatoirement un des enregistrements admissibles ayant contribué au calcul des paramètres. Notez que lorsqu'une variable de pondération est spécifiée, le poids des enregistrements est pris en considération dans le processus de sélection aléatoire. Le résidu de l'enregistrement sélectionné, qui est la différence entre la valeur reportée et la valeur estimée par le modèle de régression, est ensuite calculé. Si une variable de variance du modèle a été spécifiée lors de la définition de l'algorithme, elle sera prise en considération dans le calcul de l'erreur aléatoire. Ainsi, le terme d'erreur aléatoire ajouté à la valeur imputée y_i est donné par

$$\hat{\varepsilon}_i = R_J \sqrt{\frac{V_{iT_{m+1}}^{p_{m+1}}}{V_{jT_{m+1}}^{p_{m+1}}}}$$

où R_J est le résidu de l'enregistrement admissible J aléatoirement sélectionné et v est la variable de variance du modèle. Notez que si aucune variable de variance du modèle n'est spécifiée lors de la définition de l'algorithme, le terme d'erreur aléatoire $\hat{\varepsilon}_i$ est simplement le résidu R_J .

Calcul des paramètres d'une régression linéaire

Comme mentionné précédemment, la forme générale du modèle de régression utilisé pour représenter les estimateurs par régression linéaire est

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1T_1}^{p_1} + \hat{\beta}_2 x_{i2T_2}^{p_2} + \hat{\beta}_3 x_{i3T_3}^{p_3} + \cdots + \hat{\beta}_m x_{imT_m}^{p_m} + \hat{\varepsilon}_i$$

où les T_j font référence à des périodes courantes ou historiques et les p_j sont des exposants. La variable à imputer y_i est la variable dépendante du modèle alors que les variables auxiliaires x_{ij} sont les variables indépendantes, ou régresseurs.

Les paramètres d'un algorithme de régression linéaire sont les coefficients de régression, notés $\hat{\beta}_j$. Les valeurs de ces paramètres sont calculées à l'aide de la méthode des moindres carrés. Supposons que les valeurs des variables indépendantes x_{ij} , élevées à l'exposant approprié, sont représentées par la matrice X suivante :

$$X = \begin{pmatrix} 1 & x_{11T_1}^{p_1} & x_{12T_2}^{p_2} & x_{13T_3}^{p_3} & \cdots & x_{1mT_m}^{p_m} \\ 1 & x_{21T_1}^{p_1} & x_{22T_2}^{p_2} & x_{23T_3}^{p_3} & \cdots & x_{2mT_m}^{p_m} \\ 1 & x_{31T_1}^{p_1} & x_{32T_2}^{p_2} & x_{33T_3}^{p_3} & \cdots & x_{3mT_m}^{p_m} \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ 1 & x_{n1T_1}^{p_1} & x_{n2T_2}^{p_2} & x_{n3T_3}^{p_3} & \cdots & x_{nmT_m}^{p_m} \end{pmatrix}$$

La h^e ligne de X représente le h^e enregistrement admissible pour le calcul des coefficients de régression, et le nombre de lignes de X est le nombre d'enregistrements admissibles. La première colonne de X est associée avec le paramètre d'ordonnée à l'origine $\hat{\beta}_0$. Si l'ordonnée à l'origine n'est pas incluse dans le modèle de régression, alors cette première colonne est exclue de la matrice X . Les autres m colonnes de X sont associées avec les variables auxiliaires indépendantes x_{ij} . Il y a autant de ces colonnes dans X qu'il y a de variables indépendantes dans le modèle de

régression. Les valeurs courantes de la variable à imputer pour les enregistrements admissibles sont représentées dans le vecteur colonne \mathbf{Y} suivant :

$$\mathbf{Y} = \begin{pmatrix} y_{1C} \\ y_{2C} \\ \vdots \\ y_{nC} \end{pmatrix}$$

Si une variable de variance du modèle v a été spécifiée dans la définition de l'algorithme utilisé, ou si une variable de pondération w est incluse dans la spécification de l'opération d'imputation, elles sont prises en considération dans la matrice diagonale \mathbf{D} qui suit :

$$\mathbf{D} = \begin{pmatrix} \frac{w_{1C}}{v_{1T_{m+1}}^{p_{m+1}}} & 0 & 0 & \dots & 0 \\ 0 & \frac{w_{2C}}{v_{2T_{m+1}}^{p_{m+1}}} & 0 & \dots & 0 \\ 0 & 0 & \frac{w_{3C}}{v_{3T_{m+1}}^{p_{m+1}}} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{w_{nC}}{v_{nT_{m+1}}^{p_{m+1}}} \end{pmatrix}$$

Notez que si aucune variable de pondération w n'est spécifiée, alors w_{hC} prend la valeur 1 pour tous les enregistrements admissibles. Dans le même ordre d'idée, si aucune variable de variance du modèle n'est spécifiée, alors $v_{hT_{m+1}}^{p_{m+1}}$ prend la valeur 1 pour tous les enregistrements admissibles.

Le vecteur colonne des coefficients de régression est dénoté par

$$\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_m \end{pmatrix}$$

si une ordonnée à l'origine est spécifiée, $\hat{\beta} = \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \\ \vdots \\ \hat{\beta}_m \end{pmatrix}$ sinon

Les valeurs des coefficients de régression sont obtenues en résolvant le système linéaire suivant :

$$(\mathbf{X}'\mathbf{D}\mathbf{X})\hat{\beta} = \mathbf{X}'\mathbf{D}\mathbf{Y}$$

La solution est donnée par

$$\hat{\beta} = (\mathbf{X}'\mathbf{D}\mathbf{X})^{-1}\mathbf{X}'\mathbf{D}\mathbf{Y}$$

Exclusion dans le calcul des paramètres

Lorsque des paramètres sont requis par des algorithmes, ils sont calculés pour chaque exécution de la procédure Proc Estimator, avant que l'imputation proprement dite soit effectuée. Tous les algorithmes spécifiés pour la procédure utilisent ensuite ces paramètres calculés; les valeurs des paramètres ne changent pas pendant la procédure pour refléter les nouvelles valeurs qui viennent d'être imputées. En plus de choisir une variable de pondération, l'utilisateur peut influer sur le calcul des paramètres en choisissant d'inclure ou d'exclure trois types de champs dans le calcul des paramètres. Ces trois types de champs sont :

- les champs à exclure (FTE);
- les champs ayant déjà fait l'objet d'une imputation;
- toutes les variables qui sont indiquées par l'utilisateur qui excluent des enregistrements à l'égard des ensembles de données SAS courantes ou historiques.

En plus des enregistrements dont l'utilisateur a spécifié l'exclusion, Banff peut automatiquement exclure certains enregistrements afin que les paramètres d'un même algorithme soient calculés à partir des mêmes enregistrements. Nous allons maintenant étudier chacun de ces quatre types d'exclusion.

Le code FTE (champ à exclure), qui peut être utilisé par la procédure d'imputation par enregistrement donneur pour éviter que certaines valeurs soient transférées dans les enregistrements receveurs, peut aussi servir à indiquer à la procédure d'imputation par estimateur les enregistrements qui ne doivent pas contribuer aux paramètres des algorithmes. Les valeurs contenues dans les champs FTE ne sont pas suffisamment extrêmes pour nécessiter une imputation, mais elles sont assez exceptionnelles pour que l'utilisateur souhaite les exclure du calcul des paramètres. L'exclusion des valeurs aberrantes ne réfère qu'aux codes FTE contenus dans le champ à imputer et les variables auxiliaires; les codes FTE des autres champs sont ignorés. Le lecteur trouvera à la section 4 (Détection des valeurs aberrantes) une description indiquant comment repérer les champs FTE.

Lorsque le système calcule un paramètre, il est possible que certains champs aient déjà fait l'objet d'une imputation au moyen de la procédure d'imputation par enregistrement donneur ou d'une autre méthode d'imputation par estimateur. L'utilisateur a la possibilité d'exclure les valeurs contenues dans ces champs du calcul des paramètres afin de s'assurer que seules les valeurs initialement fournies par les répondants sont utilisées pour les fins de ce calcul. Certains utilisateurs considèrent que cette option constitue un avantage important, tandis que d'autres préfèrent calculer les paramètres à partir du plus grand nombre possible d'enregistrements. L'exclusion des valeurs imputées présente un inconvénient : le fait de calculer les paramètres uniquement à partir des données ayant satisfait aux règles de vérification initiales peut provoquer une certaine distorsion des paramètres. Par exemple, supposons qu'un fort pourcentage d'enregistrements relatifs aux unités de grande taille ont satisfait aux règles de vérification, contre seulement un faible pourcentage des enregistrements relatifs aux unités de petite taille. Si certaines opérations d'imputation ont été effectuées et si ces opérations ont préservé la relation entre les unités de grande taille et celles de petite taille, le fait d'utiliser toutes les données disponibles devrait permettre d'obtenir des moyennes plus proches des moyennes globales « réelles ». Le fait d'exclure les valeurs contenues dans les champs ayant fait l'objet d'une imputation devrait se traduire par l'obtention de moyennes reflétant davantage le comportement des unités de grande taille. Prenez note que les valeurs dans les champs imputés par la procédure d'imputation déterministe sont traitées comme données initiales.

L'utilisateur peut aussi spécifier une « variable d'exclusion » dans les ensembles de données SAS courantes et/ou historiques qui indiquent les enregistrements qui doivent être exclus pour le calcul des paramètres. La variable d'exclusion demeure en vigueur pendant toute l'exécution de la procédure. L'utilisateur peut spécifier la même variable d'exclusion ou une autre, différente, pour l'application suivante de la procédure.

En plus des enregistrements dont l'utilisateur a spécifié l'exclusion, Banff peut automatiquement exclure certains champs du calcul des paramètres afin que les coefficients de régressions, ou les moyennes figurant au numérateur et au dénominateur des fonctions d'estimation, soient calculés à partir du même ensemble d'enregistrements. Cette exclusion réduit encore davantage le nombre de valeurs déclarées par les répondants dont on dispose pour le calcul des paramètres.

Critères relatifs au calcul des paramètres

Si l'algorithme nécessite le calcul de paramètres, l'utilisateur spécifie obligatoirement les nombre et pourcentage minimaux d'enregistrements admissibles dont doit disposer le système pour procéder au calcul des paramètres. Comme ces critères sont vérifiés après que toutes les exclusions aient été effectuées, l'utilisateur doit tenir compte du nombre prévu d'exclusions au moment de spécifier le nombre minimal et le pourcentage minimal d'enregistrements admissibles requis. Si les critères ne sont pas respectés, Banff ne procédera pas au calcul des paramètres et il imprimera un message à cet effet.

Calcul des estimateurs utilisés pour l'imputation – Première exécution de la procédure Proc Estimator

Prenons l'exemple suivant.

Supposons que l'utilisateur ait décidé d'utiliser comme première opération du groupe une régression linéaire non pondérée de la forme : $y_{iC} = \beta_0 + \beta_1 x_{iC}$. Notez que cette régression est la même que l'algorithme pré-défini CURREG. Si cet estimateur n'est pas disponible comme algorithme pré-défini par le système, l'utilisateur doit d'abord le définir comme suit :

Algorithme :	CURREG
Type :	LR
État :	LR1
Format :	Intercept, aux1(c)
Description :	Une régression linéaire simple basée sur une variable indépendante provenant du fichier de données courantes.

Une fois que l'utilisateur a défini le nouvel algorithme, il peut procéder à la spécification de l'estimateur. Notez qu'aucune variable de pondération n'est spécifiée et que la variable d'exclusion est présente dans le fichier de données courantes. L'utilisateur a choisi d'ajouter un terme d'erreur aléatoire à la valeur imputée.

Champ à imputer :	y	Exclude valeurs aberrantes (Y/N) :	Y
Algorithme :	CURREG	Exclude Imputed (Y/N) :	Y
Variable auxiliaire :	x	Exclude enregistrements du	
Terme d'erreur (Y/N) :	Y	fichier courant pour lesquels :	z = 0
		Accepter valeurs négatives (Y/N) :	N

Ident	Fichier courant - Avant imputation				
	w	x	y	z	EXCL
R01	10	99 FTE	4	200	
R02	10	4	- 1 FTI	150	
R03	10	4	7	250	
R04	10	-2	9 IDN	200	
R05	10	2	5	0	E
R06	10	3 FTI	8	100	
R07	5	6	12	500	
R09	5	6	14	600	
R10	5	-1	- 1 FTI	500	

Supposons que la variable y doit être imputée au moyen de l'algorithme CURREG en utilisant x comme variable auxiliaire. Comme l'utilisateur a choisi d'exclure les champs FTE, les valeurs imputées et aussi les enregistrements du fichier de données courantes pour lesquels z = 0. En plus, l'utilisateur a choisi de ne pas accepter les valeurs négatives. Plusieurs enregistrements ne seront donc pas utilisés pour le calcul des paramètres, en l'occurrence les coefficients de régression. Puisque le calcul des coefficients de cette régression linéaire fait intervenir à la fois les champs x et y du fichier courant, seuls les enregistrements contenant des données valides pour ces deux champs seront retenus.

Dans le présent exemple, l'enregistrement R01 ne peut être utilisé car la variable x s'est vue attribuer le code FTE. L'enregistrement R02 n'est pas admissible parce que le champ contenant la variable y doit être imputé. L'enregistrement R03 est admissible. L'enregistrement R04 ne peut être utilisé parce qu'un de ses champs (y) a été imputé au moyen de la méthode d'imputation par enregistrement donneur. En plus, il y a une valeur négative pour la variable x. L'enregistrement R05 n'est pas admissible parce que la variable z prend une valeur satisfaisant l'expression d'exclusion (variable d'exclusion EXCL = 'E'), alors les deux valeurs x et y de cet enregistrement sont également exclues du calcul des paramètres. La variable x a « FTI » pour code d'état dans l'enregistrement R06, donc R06 ne peut pas contribuer au calcul des paramètres. Les enregistrements R07 et R09 sont admissibles. Enfin, l'enregistrement R10 ne peut être utilisé parce que la valeur prise par la variable y doit être imputée. Veuillez également noter que l'on ne peut pas imputer la variable y pour R10 à cette étape-ci, car l'algorithme a besoin de disposer d'une valeur valide pour la variable auxiliaire x, mais une valeur négative n'est pas acceptable.

Avec le modèle de régression spécifié, la solution de la méthode des moindres carrés pour les coefficients de régression est :

$$\hat{\beta}_0 = \bar{y}_C - \hat{\beta}_1 \bar{x}_C, \quad \hat{\beta}_1 = \frac{\sum (x_{iC} - \bar{x}_C)(y_{iC} - \bar{y}_C)}{\sum (x_{iC} - \bar{x}_C)^2}, \quad \hat{\varepsilon}_i = y_{JC} - \hat{\beta}_0 - \hat{\beta}_1 x_{JC}$$

Les moyennes non pondérées des valeurs courantes prises par y (la variable à imputer) et x (la variable auxiliaire) sont calculées à partir des trois enregistrements admissibles comme suit :

$$\bar{y}_C = \frac{7 + 12 + 14}{3} = \frac{33}{3} = 11$$

$$\bar{x}_C = \frac{4 + 6 + 6}{3} = \frac{16}{3} = 5 \frac{1}{3}$$

Ainsi, les valeurs calculées des paramètres sont $\hat{\beta}_0 = -5$ et $\hat{\beta}_1 = 3$. Supposons que Banff ait aléatoirement choisi R07 comme l'enregistrement ayant à donner son résidu à l'enregistrement R02. Ainsi, l'erreur aléatoire pour R02 serait :

$$\hat{\epsilon}_{R02} = \hat{\epsilon}_{R07} = 12 - (-5) - (3 \times 6) = -1$$

Finalement, la valeur de y devant être imputée dans l'enregistrement R02 au moyen de l'algorithme CURREG serait :

$$\hat{y}_{R02C} = \hat{\beta}_0 + \hat{\beta}_1 x_{R02C} + \hat{\epsilon}_{R02} = -5 + (3 \times 4) - 1 = 6$$

Calcul des estimateurs utilisés pour l'imputation – Deuxième exécution de la procédure Proc Estimator, premier estimateur

Supposons maintenant que l'utilisateur demande l'imputation à l'aide de la méthode de la tendance des différences (DIFTREND), lors d'une deuxième exécution de la procédure Proc Estimator. L'utilisateur a décidé d'employer une variable de pondération, et de ne pas ajouter un terme d'erreur aléatoire à la valeur imputée. En plus, les valeurs négatives sont valides. Une variable auxiliaire n'est pas requise pour cet algorithme. Les enregistrements contenant $z = 0$ dans le fichier de données courantes seront exclus. Une fois de plus, l'utilisateur a choisi d'exclure les valeurs aberrantes mais va permettre aux valeurs imputées de contribuer aux paramètres. Notez que la valeur prise par y dans l'enregistrement R02 est une valeur qui a été imputée à l'aide de l'algorithme CURREG lors de la première exécution de la méthode. Le code ILR1 (imputation par CURREG) indique la méthode d'imputation utilisée.

Champ à imputer :	y	Exclure valeurs imputées (Y/N) :	N
Algorithme :	DIFTREND	Exclure enregistrements du	
Variable de pondération :	w	fichier courant pour lesquels :	$z = 0$
Terme d'erreur (Y/N) :	N	Accepter valeurs négatives (Y/N) :	Y
Exclure valeurs aberrantes (Y/N) :	Y		

Ident	Fichier courant					EXCL	Fichier historique				
	w	x	y	z			Ident	w	x	y	z
R01	10	99	FTE	4	200		R01	8	3	6	125
R02	10	4		6	ILR1	150	R02	8	2	7	100
R03	10	4		7		250	R04	8	2	4	0
R04	10	-2		9	IDN	200	R05	8	1	4	150
R05	10	2		5		0	R06	8	2	7	300
R06	10	3	FTI	8		100	R07	4	7	12	200
R07	5	6		12		500	R08	4	4	8	175
R09	5	6		14		600	R09	4	5	10	200
R10	5	-1		-1	FTI	500	R10	4	-7	14	250

Dans cet exemple, le système doit calculer deux moyennes : y dans le fichier de données courantes et y dans le fichier de données historiques. Pour cet algorithme, si un enregistrement ne contribue pas à une moyenne, alors il ne peut pas aussi contribuer à un autre puisque les moyennes apparaissant au numérateur et au dénominateur de l'estimateur doivent être basées sur les mêmes enregistrements.

Dans ce cas-ci, l'enregistrement R01 est utilisé puisque le code FTE n'a pas été attribué à une des variables dont il faut calculer la moyenne. L'utilisateur a choisi d'inclure les données imputées dans le calcul des moyennes donc l'enregistrement R02 est admissible puisque la valeur de y pour R02 a été imputée à l'exécution précédente de la procédure Proc Estimator. L'enregistrement R03 n'est pas utilisé parce qu'il n'existe pas d'enregistrement historique correspondant, et que le numérateur et le dénominateur doivent être calculés à partir des mêmes enregistrements. L'enregistrement R04 est admissible parce que l'utilisateur a permis l'utilisation des données imputées dans le calcul des paramètres, et que la valeur de y a été imputée par la méthode d'imputation par donneur avant la présente exécution de Proc Estimator. On notera que la valeur de z pour R04 dans le fichier historique est z = 0, mais la condition d'exclusion spécifie seulement z = 0 dans le fichier de données courantes. L'enregistrement R05 n'est pas utilisé parce que la valeur prise par z dans le fichier courant satisfait à l'expression d'exclusion (variable d'exclusion EXCL = 'E'). L'enregistrement R08 n'est pas utilisé parce qu'il n'existe pas d'enregistrement courant correspondant. Les enregistrements R06, R07 et R09 sont utilisés. Enfin, l'enregistrement R10 n'est pas utilisé parce que la variable y s'est vue attribuer le code FTI (champ à imputer).

Les moyennes des valeurs courantes et historiques prises par y, la variable à imputer, dans les cinq enregistrements admissibles sont calculées comme suit, en utilisant la variable de pondération w :

$$\bar{y}_C = \frac{(10 \times 4) + (10 \times 6) + (10 \times 9) + (10 \times 8) + (5 \times 12) + (5 \times 14)}{10 + 10 + 10 + 10 + 5 + 5} = \frac{400}{50} = 8$$

$$\bar{y}_H = \frac{(8 \times 6) + (8 \times 7) + (8 \times 4) + (8 \times 7) + (4 \times 12) + (4 \times 10)}{8 + 8 + 8 + 8 + 4 + 4} = \frac{280}{40} = 7$$

La valeur de y devant être imputée pour l'enregistrement R10 au moyen de la méthode d'imputation par la tendance des différences serait :

$$\hat{y}_{R10C} = \frac{\bar{y}_C}{\bar{y}_H} y_{R10H} = \frac{8}{7} \times 14 = 16.$$

Calcul des estimateurs utilisés pour l'imputation – Deuxième exécution du Proc Estimator, deuxième estimateur

Supposons maintenant que pour la deuxième exécution de Proc Estimator, l'utilisateur décide de procéder à l'imputation d'une autre variable, en l'occurrence l'imputation de x au moyen de l'algorithme pré-défini du quotient des valeurs courantes (CURRATIO). L'utilisateur doit réaliser que tous les estimateurs dans une même exécution de la méthode Proc Estimator doivent partager la même condition d'exclusion. Comme il ne s'agit pas d'une nouvelle exclusion de la méthode, l'utilisateur doit employer la même condition d'exclusion que pour l'imputation de y avec l'estimateur DIFTREND. Dans cet exemple, l'utilisateur a de nouveau choisi d'exclure les enregistrements du fichier de données courantes pour lesquels z = 0. L'utilisateur a également décidé d'inclure une variable de pondération, et d'exclure les valeurs aberrantes mais a décidé d'utiliser les valeurs déjà imputées afin qu'elles contribuent aux paramètres. Les valeurs négatives sont encore valides. Un terme d'erreur aléatoire sera ajouté aux valeurs imputées. On notera que la valeur prise par y dans l'enregistrement R10 a été imputée à l'aide de l'estimateur Difference Trend à la deuxième (courante) exécution de la procédure Proc Estimator. Le code IDT (imputation par DIFTREND) indique la méthode d'imputation utilisée.

Champ à imputer :	x	Exclure valeurs aberrantes (Y/N) :	Y
Algorithme :	CURRATIO	Exclure valeurs imputées (Y/N) :	N
Algorithme :	y	Exclure enregistrements du	
Variable de pondération :	w	fichier courant pour lesquels :	$z = 0$
Terme d'erreur (Y/N) :	Y	Accepter valeurs négatives (Y/N) :	Y

Fichier courant						Fichier historique					
Ident	w	x	y	z	EXCL	Ident	w	x	y	z	
R01	10	99	FTE	4		R01	8	3	6	125	
R02	10	4		6	ILR1	150	R02	8	2	7	100
R03	10	4		7			R04	8	2	4	0
R04	10	-2		9	IDN	200	R05	8	1	4	150
R05	10	2		5			R06	8	2	7	300
R06	10	3	FTI	8			R07	4	7	12	200
R07	5	6		12			R08	4	4	8	175
R09	5	6		14			R09	4	5	10	200
R10	5	-1		16	IDT	500	R10	4	-7	14	250

Dans cet exemple, le système doit calculer deux moyennes : x dans le fichier de données courantes et y dans le fichier de données courantes. Comme pour l'exemple précédent, si un enregistrement ne contribue pas à une moyenne, alors il ne peut aussi contribuer à une puisque que les moyennes apparaissant au numérateur et au dénominateur de l'estimateur doivent être basées sur les mêmes enregistrements.

L'enregistrement R01 ne peut pas être utilisé parce que la variable x s'est vue attribuer le code d'état FTE dans le fichier courant. L'enregistrement R02 est admissible parce que l'utilisateur a choisi d'inclure les données imputées dans le calcul des paramètres à cette étape et que la valeur de y pour R02 a été imputée à l'exécution précédente de la procédure Proc Estimator. L'enregistrement R03 est utilisé. L'enregistrement R04 est admissible, parce que sa valeur courante pour y a été imputée précédemment selon la procédure d'imputation par enregistrement donneur. On ne peut pas utiliser l'enregistrement R05, car le fichier courant contient la valeur z = 0 (ce qui répond à la variable d'exclusion EXCL = 'E'). L'enregistrement R06 n'est pas utilisé parce que sa valeur courante pour la variable x a pour code d'état « FTI ». L'enregistrement R08 est inadmissible, parce qu'il n'y a pas d'enregistrement courant correspondant. Les enregistrements R07 et R09 sont utilisés. L'enregistrement R10 n'est pas utilisé car sa valeur courante pour la variable y a été imputée pendant l'exécution active (la deuxième) de la procédure. En d'autres mots, toutes les valeurs des paramètres sont calculées au début de l'exécution de la procédure, avant même qu'une imputation soit effectuée. Seules les valeurs imputées à l'exécution précédente de la procédure seront prises en compte. Les paramètres ne sont pas recalculés pour tenir compte des données qui ont été imputées pendant l'exécution active de la procédure.

Les moyennes des valeurs courantes prises par x (la variable à imputer) et y (la variable auxiliaire), à l'aide de la variable de pondération w, sont calculées comme suit dans les cinq enregistrements admissibles :

$$\bar{x}_C = \frac{(10 \times 4) + (10 \times 4) + (10 \times -2) + (5 \times 6) + (5 \times 6)}{10 + 10 + 10 + 5 + 5} = \frac{120}{40} = 3$$

$$\bar{y}_C = \frac{(10 \times 6) + (10 \times 7) + (10 \times 9) + (5 \times 12) + (5 \times 14)}{10 + 10 + 10 + 5 + 5} = \frac{350}{40} = 8.75$$

Supposons que Banff ait aléatoirement choisi R03 comme l'enregistrement ayant à donner son résidu à l'enregistrement R06. Ainsi, l'erreur aléatoire pour R06 serait :

$$\hat{\varepsilon}_{R06} = \hat{\varepsilon}_{R03} = x_{R03C} - \frac{\bar{x}_C}{\bar{y}_C} y_{R03C} = 4 - \frac{3}{8.75} \times 7 = 1.6$$

Ainsi, la valeur de x à être imputée pour l'enregistrement R06 à l'aide de l'estimateur par le quotient des valeurs courantes avec une terme d'erreur aléatoire serait :

$$x_{R06C} = \frac{\bar{x}_C}{\bar{y}_C} y_{R06C} + \hat{\varepsilon}_{R06} = \frac{3}{8.75} \times 8 + 1.6 = 4.34$$

9. PROC PRORATE – AJUSTEMENT AU PRORATA

Objet

Cette procédure permet d'ajuster au prorata chaque enregistrement selon une règle de vérification d'égalité afin de s'assurer que la somme des composantes est égale au total fixe. Elle possède des fonctions qui lui permettent de faire la distinction entre des données imputées antérieurement et des données originales, donc d'être exécuté à titre de procédure post-imputation ou de procédure isolée. Dans tout cas où la somme des parties n'est pas égale au total, l'inégalité en question est soumise à un des deux algorithmes d'ajustement au prorata disponibles afin d'établir, par transformation proportionnelle, l'égalité entre les composantes et le total. Puis, l'équation est soumise à un algorithme d'arrondissement pour s'assurer que les valeurs de sortie des variables concernées contiennent le nombre approprié de décimales.

Description de la méthode employée

L'utilisateur fournit une série de règles de vérification d'égalité dont les valeurs doivent être ajustées au prorata qui fait partie d'un groupe de règles de vérification. Seuls des opérateurs d'addition peuvent être utilisés dans les sommes. Les règles de vérification peuvent être emboîtées, c'est-à-dire dépendantes, dans un groupe de règles de vérification, mais elles ne peuvent pas être indépendantes. Les variables ne peuvent figurer qu'une seule fois dans le premier membre de l'équation (d'une somme) et, par conséquent, qu'une fois dans le second membre (sous forme de total fixé). Exemple :

Groupe de vérification 1

$$\begin{aligned} \text{tot1} + \text{tot2} &= \text{grandtotal} \\ \text{a} + \text{b} + \text{c} &= \text{tot1} \\ \text{d} + \text{e} + \text{f} &= \text{tot2} \end{aligned}$$

L'utilisateur peut spécifier les règles dans n'importe quel ordre, parce que Banff a un parseur pour organiser les règles dans les groupes distincts et l'ordre approprié. Noter que pour le groupe dans l'exemple, les deuxième et troisième règles forment un sous-groupe du premier groupe.

Les règles sont appliquées à chaque enregistrement dans un ordre hiérarchique et si les inégalités ne sont pas satisfaites, les algorithmes de prorata et d'arrondissement sont faits. Lorsque les éléments d'une équation sont ajustés, le total à la droite du signe d'égalité ne change pas.

L'utilisateur peut aussi préciser quelles variables individuelles seront soumises à l'ajustement au prorata, enregistrement par enregistrement, en indiquant si seules des variables dont les valeurs ont été imputées antérieurement, seules les données originales ou à la fois les données imputées et originales sont admissibles. Pour mettre cette fonction en oeuvre, le système lit les codes des variables concernées tels qu'ils figurent dans le fichier SAS d'entrée qui contient les codes. Toute variable dont le code d'état commence par « I » (exception unique : « IDE » pour l'imputation déterministe) est considérée comme une variable imputée. Tout autre code d'état ou l'absence d'une variable dans le fichier d'entrée d'états indique au système que les valeurs de la variable sont des données originales. Les variables inadmissibles pour le prorata sont éliminées de la somme et le total est ajusté en conséquence avant que débute l'ajustement au prorata. Les valeurs de zéro sont, elles aussi, éliminées à ce stade, puisque le processus est incapable de les modifier.

L'utilisateur a aussi le choix de préciser les poids à appliquer à chaque composante visée par l'opération d'addition. Cette option permet de contrôler la variation relative de la valeur de chaque composante à la suite de l'ajustement au prorata, la variation étant inversement proportionnelle au poids appliqué. Les poids sont appliqués au niveau de la composante; par conséquent, le même poids s'applique à tous les enregistrements.

Contrôle de la syntaxe des règles de vérification

Avant de procéder à l'ajustement au prorata, le système exécute une série de contrôle des règles de vérification pour s'assurer que les règles essentielles de syntaxe sont respectées. Le système confirme notamment la structure hiérarchique des règles et la positivité des poids. Si une erreur est décelée, au lieu d'exécuter l'ajustement au prorata, le système produit le message d'erreur pertinent.

Algorithme d'ajustement au prorata

Si une somme n'est pas égale au total, par exemple $x_1 + x_2 + \dots + x_n \neq y$, le système applique un algorithme d'ajustement au prorata. Il y a deux méthodes disponibles: la méthode de **base** et la méthode « **scaling** ».

Quand toutes les composantes ont le même signe (toutes positives ou toutes négatives), les deux méthodes donnent des résultats équivalents si les valeurs des paramètres, options, poids, etc., sont aussi équivalentes.

Algorithme de la méthode de base

La valeur ajustée pour chacune des m_h variables admissibles x_{hi} dans une règle h ($h=1,\dots,r$) est calculée :

$$x'_{hi} = x_{hi} + \frac{x_{hi}}{w_{hi}} \times \frac{y_h - \sum_{j=1}^{m_h} x_{hj}}{\sum_{j=1}^{m_h} (x_{hj}/w_{hj})} \quad i = 1, 2, 3, \dots, m$$

où y_h est le total fixé pour la règle h et w_{hi} est le poids associé à x_{hi} .

L'ajustement au prorata par la méthode de base non pondérée est simplement un cas spécial de l'ajustement au prorata pondéré où tous les $w_{hi}=1$:

$$x'_{hi} = \frac{y_h}{\sum_{j=1}^{m_h} x_{hj}} x_{hi}$$

Avec la méthode de base, il est possible que le signe d'un valeur puisse changer, à moins que l'utilisateur empêche ce changement par la bonne spécification du paramètre de la limite inférieure de variation. Ce paramètre est discuté ci-dessous. De plus, si l'utilisateur empêche un changement du signe et il y a un mélange des valeurs positives et négatives, et même si le changement net doit être positive (ou négative) pour que la somme soit égale au total fixé, il est possible que quelques composantes aient les changements positifs, et les autres composantes aient les changements négatifs. Ceci dépend des signes des composantes et le signe du total.

Algorithme de la méthode « scaling »

Premièrement, le facteur k pour une règle h ($h=1, \dots, r$) est calculé pour les m_h variables x_{hi} qui sont admissibles :

$$k_h = \frac{\sum_{i=1}^{m_h} x_{hi} - y_h}{\sum_{i=1}^{m_h} |x_{hi}/w_{hi}|} \quad h = 1, 2, 3, \dots, r$$

où y_h est le total fixé pour la règle h et w_{hi} est le poids associé à x_{hi} .

Quand $-1 \leq k_h \leq 1$, la valeur ajustée de x_{hi} pour la règle h est :

$$\begin{aligned} x'_{hi} &= (1 - k_h/w_{hi})x_{hi} && \text{si } x_{hi} > 0 \\ x'_{hi} &= (1 + k_h/w_{hi})x_{hi} && \text{si } x_{hi} < 0 \end{aligned} \quad i = 1, 2, 3, \dots, m$$

Il est une circonstance extrême quand $k > 1$ ou $k < -1$, par exemple dans un cas bizarre où le total y_h est une valeur positive et tous les x_{hi} sont négatifs. Cette situation indique qu'un problème extrême existe et l'utilisateur devrait aborder ce problème au stage de vérification des données avant le prorata. Si le calcul donne $k > 1$ ou $k < -1$, le prorata n'exécutera pas pour cet enregistrement et Banff donnera un message d'avertissement.

Sous la méthode scaling, il n'est pas possible que le signe de la valeur change. Les valeurs négatives ne peuvent pas devenir positives, et vice-versa. De plus, si le changement net est positif (négatif), tous les changements aux composantes seront positifs (négatifs).

Algorithme d'arrondissement

Lorsque l'application de l'algorithme d'ajustement au prorata a été effectuée, un grand nombre de décimales supplémentaires a dû être conservé pour que la somme soit égale au total. Par conséquent, le système doit arrondir les données concernées au nombre de décimales précisé par l'utilisateur, en veillant à ce que l'égalité nouvellement créée continue d'être respectée. L'algorithme d'arrondissement ajuste les valeurs qui figurent dans tous les champs au nombre correct de décimales en assurant que la somme demeure égale au total. Pour la règle h :

Première étape. Arrondir les x'_{hi} à $d+1$ décimales près; d =nombre de décimales précisée par l'utilisateur pour le groupe de règles de vérification concerné.

Deuxième étape. Pour $i=1$, arrondir x'_{h1} au chiffre supérieur ou inférieur pour obtenir x''_{h1} (avec d décimales).

Troisième étape. Pour $i>1$, arrondir u_i :

$$u_{hi} = x'_{hi} + \sum_{j=1}^{i-1} [x'_{hj} - x''_{hj}]$$

à la valeur finale ajustée au prorata x''_{h1} .

Si l'utilisateur spécifie $d = \text{nombre de décimales requises}$, trois conditions doivent toujours être satisfaites :

- d doit être inférieur ou égal au nombre de décimales qui existe dans le fichier de données;
- d doit être supérieur ou égal au nombre de décimales du total;
- d doit être compris dans l'intervalle $[0,9]$.

La valeur par défaut de d est zéro.

Après que l'ajustement et l'arrondissement sont complétés pour tous les x_{hi} , la procédure continue dans l'ordre hiérarchique pour chacun des autres $r-1$ règles. La procédure tient compte des valeurs changées pour les x_{hi} qui peuvent devenir le nouveau total fixé y_h dans une règle subséquente.

Exemple

Considérons les données qui suivent soumises à l'ajustement au prorata en utilisant la méthode scaling. Utilisons le groupe 1 (tel que précédemment décrit) et supposons que l'utilisateur demande que toutes les variables ajustées au prorata soient arrondies à l'unité (zéro décimale) et qu'il spécifie comme suit les poids, ainsi que les catégories de variables à imputer :

Groupe 1	Poids	Variables à ajuster:
tot1 + tot2 = grandtotal	1,1	Toutes
a + b + c = tot1	1,1,2	Données imputées uniquement
d + e + f = tot2	1,1,1	Toutes

Par exemple, la variable « c » (poids =2) subirait un ajustement au prorata relatif (si nécessaire) DEUX FOIS MOINS important que les variables « a » (poids=1) et « b » (poids=1), parce que le poids qui lui est appliqué est deux fois plus grand.

L'utilisateur a également précisé que pour la première règle, toutes les variables peuvent être soumises à l'ajustement au prorata, tandis que pour la deuxième règle, seules les variables pour lesquelles il est précisé qu'elles contiennent des données déjà imputées sont admissibles et, pour la troisième règle, toutes les variables sont encore admissibles. Ces états sont décrits en détail plus bas dans le fichier d'états des données d'entrée. Les données sont aussi décrites ci-dessous. L'utilisateur a précisé que les données négatives sont valides.

Fichier des données d'entrée :

Key_Value	A	B	C	D	E	F	TOT1	TOT2	GRANDTOTAL
REC001	6	4	-4	10	9	9	12	24	31

Fichier d'états des données d'entrée :

Key_Value	FIELDID	STATUS
REC001	A	IDN
REC001	C	IDT
REC001	E	IMP

Grâce à ce fichier d'états des données d'entrée, l'utilisateur a identifié toutes les variables qui ont été imputées antérieurement. On se souviendra que les codes d'états (STATUS) qui commencent par « I » (sauf « IDE » pour l'imputation déterministe) indiquent, en principe, des données imputées antérieurement. La procédure d'ajustement au prorata suppose que toute autre variable contient des données originales.

Suivant l'ordre hiérarchique, pour la première règle les variables TOT1 et TOT2 sont les deux admissibles pour l'ajustement au prorata selon la spécification pour les variables à imputer et le fichier d'états. Donc, la procédure calcule le facteur k pour les deux composantes dans la première règle avec les poids (les deux sont égales à 1) comme :

$$k = \left(\sum_{j=1}^2 x_j - y \right) / \left(\sum_{j=1}^2 |x_j/w_j| \right) = (12 + 24 - 31) / (12 + 24) = 5/36.$$

Prochainement, les valeurs ajustées pour TOT1 et TOT2 sont :

$$TOT1' = (1 - k / w_{TOT1})(TOT1) = (1 - 5/36)(12) = 10.33$$

$$TOT2' = (1 - k / w_{TOT2})(TOT2) = (1 - 5/36)(24) = 20.67$$

Puis, la procédure applique l'algorithme d'arrondissement, qui produit :

TOT1	TOT2	GRANDTOTAL
10	21	31

En poursuivant l'ajustement au prorata pour les deuxième et troisième règles, nous obtenons :

Key_Value	A	B	C	TOT1	D	E	F	TOT2
REC001	9	4	-3	10	7.5	6.75	6.75	21

Par conséquent, à cause de la pondération, la valeur de la variable « a » a augmenté de 50 % de sa valeur originale, c'est-à-dire deux fois plus que celle de la variable « c », dont la valeur a augmenté de 25 % de sa valeur originale. La variable « b », dont la valeur n'avait pas été imputée antérieurement, n'a pu être soumise à l'ajustement au prorata pour l'enregistrement en question, et donc elle n'a pas contribué au calcul du facteur k. Toutes les variables d, e, et f ont contribué au calcul du facteur k pour la troisième règle parce que l'utilisateur a spécifié que n'importe quelles variables ont été admissibles.

Enfin, l'exécution de l'algorithme d'arrondissement donne :

Key_Value	A	B	C	TOT1	D	E	F	TOT2
REC001	9	4	-3	10	8	6	7	21

Ordre des variables dans les règles de vérification

Bien que l'utilisateur puisse spécifier les règles de vérification dans n'importe quel ordre, les différentes façons d'ordonner des variables dans une règle de vérification peuvent produire des résultats différents après avoir appliqué l'algorithme d'ajustement au prorata. Quelquefois l'excédent du total ne peut être distribué en parts égales parmi les variables qui composent la somme alors, seules quelques variables reçoivent une partie de cette différence. Dans ce cas, un changement dans l'ordre des variables peut entraîner un changement des variables qui reçoivent cette différence répartie. Par exemple, en utilisant la règle de vérification $x + y = z$, la différence répartie est 1 pour la somme suivante à être ajustée au prorata : $1 + 1 = 3$. Dans cet exemple, si le paramètre DECIMAL=0 alors seulement la première variable x recevra la différence et la somme ajustée deviendra $2 + 1 = 3$. Si cette règle de vérification est changée à $y + x = z$, la variable y sera augmentée de 1. Toujours avec cet exemple, si la première variable a un poids supérieur à la deuxième variable, la deuxième variable recevra la différence car la variation est inversement proportionnelle au poids appliquée alors, la somme ajustée deviendra $1 + 2 = 3$. En conclusion, l'ordre des variables peut avoir un effet sur les résultats d'ajustement au prorata.

Limites de variation

L'utilisateur a également l'option de contrôler la variation relative de la valeur des variables. Cette contrainte est exprimée sous forme de rapport limite supérieur et de rapport limite inférieur de variation. Par exemple, si l'utilisateur souhaite que la valeur d'aucune variable n'augmente de plus de 25 % de sa valeur originale après l'ajustement au prorata et l'arrondissement, il fixera le rapport limite supérieur à 1.25. Il est possible que la procédure calcule une valeur ajustée de zéro si le rapport limite inférieur est fixé à zéro. Pour la méthode de base, si l'utilisateur souhaite que la valeur d'aucune variable ne change de signe à la suite de l'ajustement au prorata, il peut fixer le rapport limite inférieur à zéro ou plus que zéro. Pour la méthode scaling, par défaut le rapport limite inférieur doit être au moins zéro, afin d'éviter un changement de signe. Ces limites de variation sont appliquées après l'exécution de l'algorithme d'arrondissement. À ce stade, si le système constate que la variation de la valeur d'un champ dépasse la limite supérieure ou inférieure, la procédure d'ajustement au prorata ne traitera plus aucune autre donnée de l'enregistrement en question et passera à l'enregistrement suivant. La procédure affiche un message pour avertir l'utilisateur qu'une limite de variation a été dépassée, en précisant la variable concernée.

En ce qui concerne l'exemple précédent, si l'utilisateur avait fixé un rapport limite supérieur de 1.25, après l'arrondissement la procédure aurait constaté que la variable « a » (ajustée au prorata de 6 à 9) a varié de plus de 125 %. Par conséquent, l'enregistrement REC001 serait exclu de toute autre analyse et le système produirait un message expliquant en détail pourquoi l'ajustement au prorata a été infructueux pour l'enregistrement en question.

10. PROC MASSIMPUTATION –IMPUTATION MASSIVE

Objet

Pour des raisons opérationnelles, dans le cas de certaines enquêtes, on ne recueille des renseignements détaillés que pour un sous-échantillon (ou échantillon de deuxième phase) d'unités sélectionnées au hasard à partir d'un grand échantillon de première phase. La méthode classique de production d'estimations pour le sous-échantillon nécessite le calcul de poids de sous-échantillonnage, calcul qui peut être assez complexe. Une autre méthode, appelée « imputation massive », consiste à créer un fichier rectangulaire complet pour l'ensemble des unités d'échantillonnage de première phase en imputant par enregistrement donneur les valeurs manquantes pour les unités non échantillonnées, après que la vérification et l'imputation des unités d'échantillonnage de deuxième phase ont été complétées. Une procédure a été mise au point dans Banff pour faciliter cette opération, selon la méthode du voisin le plus proche. Dans le cas type de vérification et d'imputation, l'objectif est de repérer, pour un enregistrement particulier, les valeurs incorrectes, manquantes, incohérentes ou aberrantes, en supposant que le profil de rejet à la vérification est différent pour chaque enregistrement. Par contre, dans le cas de l'imputation massive, on sait quels enregistrements nécessitent une imputation; en outre, les champs à être imputées sont connus et identiques pour tous les enregistrements. De surcroît, on suppose que l'ensemble de données de base recueillies auprès de l'échantillon complet et les données supplémentaires recueillies auprès du sous-échantillon ont déjà été vérifiées et imputées (peut être à l'extérieur de Banff) et qu'il ne faut appliquer aucune règle en vue de vérifier la cohérence des sections individuelles ni entre les deux sections.

Champs d'appariement

Puisque les enregistrements receveurs ne contiennent en principe pas de données auxquelles il faut appliquer des règles de vérification, le concept consistant à utiliser des règles de vérifications à l'intérieur de groupes pour que Banff repère les champs erronés ne s'applique pas. Par conséquent, le système ne peut pas procéder à la détermination des champs d'appariement. Cependant, l'utilisateur peut spécifier les champs d'appariement servant à trouver des enregistrements donneurs. Les champs d'appariement sont spécifiés en tant que paramètres du programme.

Les données pour les champs d'appariement spécifiés par l'utilisateur doivent être dans le fichier de données d'entrée. Si une valeur d'un champ d'appariement est manquante pour un enregistrement receveur, le champ d'appariement de cet enregistrement ne peut être utilisé, puisqu'il n'existe aucune valeur valide pour le calcul de la distance. S'il n'existe aucun champ d'appariement valide, l'utilisateur peut demander au système de sélectionner au hasard un enregistrement donneur approprié pour l'enregistrement receveur.

Si l'utilisateur spécifie les champs d'appariement, le système transforme les valeurs de ces champs pour les enregistrements donneurs et receveurs pour lesquels il existe des valeurs valides. Cette étape est la même que dans la procédure d'imputation par enregistrement donneur; consulter la section 7.3.

Si l'utilisateur ne spécifie aucun champ d'appariement, le système sélectionne automatiquement un enregistrement donneur au hasard pour chaque enregistrement receveur.

Autres paramètres

Comme dans le cas de la procédure d'imputation par enregistrement donneur, l'utilisateur doit spécifier tant le pourcentage que le nombre minimal d'enregistrements donneurs nécessaires pour que l'imputation puisse avoir lieu. Si l'utilisateur ne spécifie pas la valeur de ces paramètres, le système utilise les valeurs par défaut. L'utilisateur peut aussi limiter le nombre de fois qu'un donneur est utilisé pour l'imputation. Ce nombre est illimité par défaut.

Description de la méthode employée

L'exécution du programme est presque identique à celle de la procédure d'imputation par enregistrement donneur; consulter la section 7.4. Il existe toutefois une différence significative. Il n'est pas nécessaire de procéder à des vérifications post-imputation. En fait, la procédure imputera simplement à l'enregistrement receveur les données qui figurent dans le fichier de données d'entrée après avoir repéré l'enregistrement IMPUTATION MASSIVE 10 – 2 donneur le plus proche, ou à partir du premier enregistrement donneur sélectionné au hasard si aucun champ d'appariement valide n'est disponible. Donc, il est très important que l'utilisateur fournisse des données « épurées » pour les enregistrements donneurs.

Aucun fichier de status n'est créé par Massimputation. Cependant, un statut pour l'imputation massive peut être requis par les utilisateurs dans le but de calculer la variance due à l'imputation. Il y a une option avec le Processeur Banff pour ajouter le statut IMAS au fichier global de statuts et ce, pour les champs imputés par Proc Massimputation. Pour les utilisateurs de Banff dans SAS, le statut IMAS doit être ajouté manuellement. Veuillez contacter l'équipe de soutien Banff si vous souhaitez obtenir le code SAS qui génère le statut IMAS à partir des résultats obtenus avec Proc Massimputation.

11. RÉFÉRENCES

- Chernikova, N.V. (1964). Algorithm for finding a general formula for the nonnegative solutions of a system of linear equations. **U.S.S.R. Computational Mathematics and Mathematical Physics** **4**, 151-158.
- Chernikova, N.V. (1965). Algorithm for finding a general formula for the nonnegative solution of a system of linear inequalities. **U.S.S.R. Computational Mathematics and Mathematical Physics** **5**, 228-233.
- Équipe de soutien de Banff (2017). Guide de l'usager des procédures de Banff 2.07. Rapport technique de Statistique Canada.
- Équipe de soutien de Banff (2017). Guide de l'usager du processeur Banff 2.07. Rapport technique de Statistique Canada.
- Équipe de soutien de Banff (2014). Tutoriel Banff 2.06. Rapport technique de Statistique Canada.
- Équipe de soutien de Banff (2006). Spécification des règles de vérification avec des données négatives dans Banff. Rapport technique de Statistique Canada.
- Équipe de soutien de Banff (2016). FAQ de Banff. Page intranet :
\\fld6filer\Team0167\Public\Generalized Systems\Banff\FAQ\FAQ_en_fr.pdf.
- Fellegi, I.P., et Holt D. (1976). A systematic approach to automatic edit and imputation. **Journal of the American Statistical Association** **71**, 17-35.
- Friedman, J.H., Bentley, J.L. et Finkel, R.A. (1977). An algorithm for finding best matches in logarithmic expected time. **ACM Transaction on Mathematical Software** **3**, 209-226.
- Équipe de soutien de GEIS (1991). Description des fonctions du système généralisé de vérification et d'imputation. (Révisé en octobre 2000). Rapport technique de Statistique Canada.
- Giles, P. (1989). Statistique Canada, cahier de travail de la Direction de la méthodologie n° SSMD-89-004E.
- Hidirogloiu, M.A. et Berthelot, J.-M. (1986). Contrôle statistique et imputation dans les enquêtesentreprises périodiques. **Techniques d'enquête** **12**, 79-89.
- Kovar, J.G., MacMillan, J. et Whitridge, P. (1988). Système généralisé de vérification et d'imputation - aperçu et stratégie. (Mis à jour en février 1991). Statistique Canada, cahier de travail de la Direction de la méthodologie n° BSMD-88-007E/F.
- Morabito, J. et Shields, M. (1992). Generalized Edit and Imputation System Applications User's Guide. Rapport technique de Statistique Canada, en préparation.
- Rousseeuw, Peter J. et M. Leroy, Annick (2003). Robust Regression and Outlier Detection. Wiley, New Jersey.

Rubin, D.S. (1973). Vertex generation in cardinality constrained linear programs. **Operations Research** **23**, 555-565.

Sande, G. (1978). An algorithm for the fields to impute problems of numerical and coded data. Rapport technique de Statistique Canada.

Sande, G. (1979). Numerical Edit and Imputation. Présenté à la 42^e réunion de l'Institut International de Statistique, Manille, (Philippines).

Schiopu-Kratina, I. et Kovar, J.G. (1989). Use of Chernikova's algorithm in the Generalized Edit and Imputation System. Statistique Canada, cahier de travail de la Direction de la méthodologie n° BSMD-89-001E.

Annexe A - Calcul des médianes et des quartiles

Médianes

La **médiane** est la valeur qui divise une série d'observations en deux groupes comportant chacun 50% des observations. La moitié des valeurs sont donc inférieures à la médiane, tandis que l'autre moitié lui sont supérieures.

Lorsque la série comporte un nombre impair d'observations, la médiane correspond à l'observation centrale. Soit n observations, la médiane occupera la position $\frac{(n+1)}{2}$ dans la série d'observations en ordre.

Lorsque la série comporte un nombre pair d'observations, la médiane correspond à la moyenne des deux observations centrales. Soit n observations, la médiane sera égale à la moyenne des observations occupant les positions $\frac{n}{2}$ et $\frac{n}{2} + 1$ dans la série d'observations en ordre.

Exemples de médianes

	Observations en ordre					n	Médiane
set 1:	1	2	<u>6</u>	7	9	5	6
set 2:	42	<u>59</u>	59			3	59
set 3:	3	<u>6</u>	<u>9</u>	10		4	7.5
set 4:	2	10	<u>10</u>	<u>10</u>	12	500	6
							$= (6+9)/2$
							$= (10+10)/2$

Quartiles

Le **premier quartile** divise les observations en deux groupes tels que 25% des observations sont inférieures à la valeur du premier quartile et 75%, supérieures à cette valeur. On trouve la position du premier quartile à l'aide de l'expression $.25 * (n+1)$, où n est le nombre d'observations. Si ce nombre est un entier, le premier quartile prend la valeur de l'observation occupant ce rang. Si ce nombre n'est pas un entier, mais une valeur de la forme $w.d$, le premier quartile se situe entre l'observation w et l'observation $(w+1)$. On obtient la valeur exacte du premier quartile en faisant la somme du produit de l'observation w par $(1.-d)$ et du produit de l'observation $(w+1)$ par $.d$.

Le **troisième quartile** divise les observations en deux groupes tels que 75% des observations sont inférieures à la valeur du troisième quartile et 25%, supérieures à cette valeur. On trouve la position du troisième quartile à l'aide de l'expression $.75 * (n+1)$, où n est le nombre d'observations. Si ce nombre est un entier, le troisième quartile prend la valeur de l'observation occupant ce rang. Si ce nombre n'est pas un entier, mais une valeur de la forme $w.d$, le troisième quartile se situe entre l'observation w et l'observation $(w+1)$. On obtient la valeur exacte du troisième quartile en faisant la somme du produit de l'observation w par $(1.-d)$ et du produit de l'observation $(w+1)$ par $.d$.

Exemples de quartiles

Soit une série de 8 observations en ordre: 1, 3, 6, 7, 10, 11, 12, 18. On obtient la position du premier quartile en calculant l'expression $.25 * (n+1)$, qui est égale à $.25 * 9 = 2.25$ dans le présent exemple. Le premier quartile se situe donc au quart de la distance entre les observations occupant le deuxième et le troisième rangs. Banff calcule donc la valeur du premier quartile en faisant la somme du produit de l'observation occupant le deuxième rang par .75 et du produit de l'observation occupant le troisième rang par .25. Dans ce cas-ci, cette valeur est égale à $(.75*3) + (.25*6) = 3.75$. Il est normal que la valeur du premier quartile s'approche plus de celle de la deuxième observation (3) que de celle de la troisième (6), puisqu'on a déterminé qu'il se situait au quart de la distance entre les deux observations.

De même, on peut déterminer la position du troisième quartile en calculant l'expression $.75 * (n+1)$, qui est égale à $.75 * 9 = 6.75$ dans le présent exemple. En conséquence, le troisième quartile se situe aux trois quarts de la distance entre l'observation occupant le sixième rang et celle occupant le septième rang. Dans cet exemple, le troisième quartile serait égal à la somme du produit de la sixième observation par .25 et du produit de la septième observation par .75, soit à $(.25*11) + (.75*12) = 11.75$. Encore une fois, il est normal que la valeur du troisième quartile s'approche plus de celle de la septième observation (12) que de celle de la sixième observation (11), puisqu'on a déterminé que cette valeur se situait aux trois quarts de la distance entre la sixième et la septième observations.

La médiane de cette série d'observations est 8.5, moyenne de la quatrième et de la cinquième observations.

Annexe B - Algorithmes prédéfinis dans Banff

Algorithme : AUXTREND

Équation :

$$\hat{y}_{iC} = \frac{x_{iC}}{x_{iH}} y_{iH}$$

Type : EF

Status : AT

Format : aux1(c,v) * fieldid(h,v) / aux1(h,v)

Description : La valeur de la même unité lors du cycle d'enquête précédent, corrigée en fonction de la variation d'une variable auxiliaire, est imputée.

Algorithme : AUXTREND2

Équation :

$$\hat{y}_{iC} = \frac{y_{iH}}{2} \left(\frac{u_{iC}}{u_{iH}} + \frac{v_{iC}}{v_{iH}} \right)$$

Type : EF

Status : AT2

Format : fieldid(h,v) / 2 * (aux1(c,v)/aux1(h,v) + aux2(c,v)/aux2(h,v))

Description : Une moyenne de deux AUXTREND est imputée.

Algorithme : CURAUX

Équation : $\hat{y}_{iC} = x_{iC}$

Type : EF

Status : CA

Format : aux1(c,v)

Description : La valeur courante d'une variable auxiliaire pour la même unité est imputée.

Algorithme : CURAUXMEAN

Équation : $\hat{y}_{iC} = \bar{x}_C$

Type : EF

Status : CAM

Format : aux1(c,a)

Description : La moyenne courante d'une variable auxiliaire est imputée.

Algorithme : CURMEAN

Équation : $\hat{y}_{iC} = \bar{y}_C$

Type : EF

Status : CM

Format : fieldid(c,a)

Description : La valeur moyenne de la variable à imputer pour toutes les unités admissibles à l'occasion du cycle d'enquête courant est imputée.

Algorithme :	CURRATIO
Équation :	$\hat{y}_{iC} = \frac{\bar{y}_C}{\bar{x}_C} x_{iC}$
Type :	EF
Status :	CR
Format :	fieldid(c,a) * aux1(c,v) / aux1(c,a)
Description :	Une estimation par quotient, établie à partir des moyennes calculées pour les unités admissibles à l'occasion du cycle d'enquête courant, est imputée.
Algorithme :	CURRATIO2
Équation :	$\hat{y}_{iC} = \frac{\bar{y}_C}{2} \left(\frac{u_{iC}}{\bar{u}_C} + \frac{v_{iC}}{\bar{v}_C} \right)$
Type :	EF
Status :	CR2
Format :	fieldid(c,a)/2 * (aux1(c,v)/aux1(c,a) + aux2(c,v)/aux2(c,a))
Description :	Une moyenne de deux CURRATIO est imputée.
Algorithme :	CURREG
Équation :	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 x_{iC}$
Type :	LR
Status :	LR1
Format :	intercept, aux1(c)
Description :	Régression linéaire simple basée sur une variable indépendante provenant du cycle d'enquête courant.
Algorithme :	CURREG_E2
Équation :	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 x_{iC} + \hat{\beta}_2 x_{iC}^2$
Type :	LR
Status :	LRE
Format :	intercept, aux1(c), aux1(c) ²
Description :	Régression linéaire basée sur la valeur et le carré de la valeur d'une variable provenant du cycle d'enquête courant.
Algorithme :	CURREG2
Équation :	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 u_{iC} + \hat{\beta}_2 v_{iC}$
Type :	LR
Status :	LR2
Format :	intercept, aux1(c), aux2(c)
Description :	Régression linéaire basée sur deux variables indépendantes provenant du cycle d'enquête courant.

Algorithme :	CURREG3
Équation :	$\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 u_{iC} + \hat{\beta}_2 v_{iC} + \hat{\beta}_3 w_{iC}$
Type :	LR
Status :	LR3
Format :	intercept, aux1(c), aux2(c), aux3(c)
Description :	Régression linéaire basée sur trois variables indépendantes provenant du cycle d'enquête courant.
Algorithme :	CURSUM2
Équation :	$\hat{y}_{iC} = u_{iC} + v_{iC}$
Type :	EF
Status :	SM2
Format :	aux1 + aux2
Description :	La somme de deux variables auxiliaires provenant du cycle d'enquête courant est imputée.
Algorithme :	CURSUM3
Équation :	$\hat{y}_{iC} = u_{iC} + v_{iC} + w_{iC}$
Type :	EF
Status :	SM3
Format :	aux1 + aux2 + aux3
Description :	La somme de trois variables auxiliaires provenant du cycle d'enquête courant est imputée.
Algorithme :	CURSUM4
Équation :	$\hat{y}_{iC} = u_{iC} + v_{iC} + w_{iC} + z_{iC}$
Type :	EF
Status :	SM4
Format :	aux1 + aux2 + aux3 + aux4
Description :	La somme de quatre variables auxiliaires provenant du cycle d'enquête courant est imputée.
Algorithme :	DIFTREND
Équation :	$\hat{y}_{iC} = \frac{\bar{y}_C}{\bar{y}_H} y_{iH}$
Type :	EF
Status :	DT
Format :	fieldid(c,a) * fieldid(h,v) / fieldid(h,a)
Description :	La valeur de la même unité lors du cycle d'enquête précédent, corrigée en fonction de la variation de la moyenne de cette variable, est imputée.

Algorithme : HISTREG
 Équation : $\hat{y}_{iC} = \hat{\beta}_0 + \hat{\beta}_1 y_{iH}$
 Type : LR
 Status : HLR
 Format : intercept, fieldid(h)
 Description : Régression linéaire basée sur les valeurs prises par la variable à imputer lors du cycle d'enquête précédent.

Algorithme : PREAUX
 Équation : $\hat{y}_{iC} = x_{iH}$
 Type : EF
 Status : PA
 Format : aux1(h,v)
 Description : La valeur historique d'une variable auxiliaire pour la même unité est imputée.

Algorithme : PREAUXMEAN
 Équation : $\hat{y}_{iC} = \bar{x}_H$
 Type : EF
 Status : PAM
 Format : aux1(h,a)
 Description : La moyenne historique d'une variable auxiliaire est imputée.

Algorithme : PREMEAN
 Équation : $\hat{y}_{iC} = \bar{y}_H$
 Type : EF
 Status : PM
 Format : fieldid(h,a)
 Description : La valeur moyenne de la variable à imputer pour toutes les unités admissibles à l'occasion du cycle d'enquête précédent est imputée.

Algorithme : PREVALUE
 Équation : $\hat{y}_{iC} = y_{iH}$
 Type : EF
 Status : PV
 Format : fieldid(h,v)
 Description : La valeur de la même unité lors du cycle d'enquête précédent est imputée.