Journées de méthodologie statistique de l'Insee 2022

GESTION DU SECRET POUR LA DIFFUSION GRAND PUBLIC DE CUBES MULTIDIMENSIONNELS : UNE EXPÉRIMENTATION AU SSM AGRICULTURE

Michaël LEVI-VALENSIN (*)

(*) Service de la Statistique et de la Prospective -
Bureau des Méthodes et de l'Informatique Statistiques
michael.levi-valensin@agriculture.gouv.fr

Mots-clés : secret statistique, optimisation, automatisation

Domaine concerné : **Histoire. Confidentialité, secret, diffusion**

Résumé

Comme pour toutes les sources statistiques, la mise en ligne des données agricoles sur le site Agreste du Ministère de l'Agriculture et de l'Alimentation exige de respecter les règles de diffusibilité de l'information. Le Service de la statistique et de la prospective (SSP), à l'instar des autres services statistiques et de l'Insee, met historiquement en œuvre des méthodes dites suppressives, qui consistent à ne pas diffuser les valeurs sensibles.
Une spécificité du SSP est de publier des cubes multidimensionnels sous la forme de tableaux interactifs dans lesquels les utilisateurs peuvent choisir les variables à différents niveaux de nomenclature. Dans de tels produits, il devient alors nécessaire de considérer tous les croisements possibles et les éventuels secrets induits.
Ce travail était jusqu'alors réalisé en SAS de manière heuristique par une fonction conçue en interne qui appliquait les règles de secret induit. La stratégie du SSP pour sortir de SAS a amené à réétudier ce traitement, et ainsi à examiner les techniques actuelles.

Les méthodes suppressives pour gérer le secret statistique sont généralement réalisées, et ce depuis une vingtaine d'années, par le logiciel τ-Argus. Ce dernier permet de résoudre assez facilement un programme d'optimisation sous contraintes en supprimant un nombre minimal d'informations tout en assurant le respect des règles de secret secondaire.
Plus récemment, des fonctions sous R ont été conçues pour appliquer des algorithmes similaires et faciliter les différentes étapes de secrétisation d'un fichier. Ainsi, le package **sdcTable** développé par Statistics Austria (INS d'Autriche) regroupe un grand nombre de fonctionnalités de τ-Argus pour la pose de masque sous R.

Cette contribution est un retour d'expérience sur l'utilisation de packages R pour appliquer les méthodes suppressives sur la publication de données agricoles sous forme de cubes.

Plusieurs tests ont ainsi été réalisés sur des fichiers comme l'enquête annuelle « Exploitations forestières et scieries » (EXF-SRI) sur les sciages de bois par essence ou les Surfaces agricoles utilisées (SAU) par commune au Recensement agricole 2020.
La mise en forme des tables et la construction des fichiers hiérarchiques au format approprié sont des étapes préalables mais indispensables avant toute pose de masque. Des difficultés et problèmes techniques sont apparus lors des traitements réalisés et nous avons mis en place plusieurs règles

d'automatisation afin de faciliter à l'avenir cette étape délicate, contraignante et néanmoins indispensable.
Ces techniques ont été intégrées dans une chaîne de traitement jusqu'à la mise en ligne sur le site Agreste sous forme de tableaux dynamiques.

Abstract

Statistical Disclosure Control (SDC) for tabular data is usually (at least in French statistics organisations) managed by suppressive methods. Tools have been developped for facilitating SDC methods, such as the well known software τ-Argus and more recently, R packages. Different methods have been promoted by Eurostat, based on pertubation principles rather than cell content masking. The statistical department of the Ministry of Agriculture, has recently re-engineered its SDC methodology and adopted the sdcTable package for the multidimensional datasets published on its website. This paper is a sharing of experience with its use.

1. Des méthodes suppressives et des logiciels

1.1. Rappel du contexte

Le secret statistique correspond à la mise en œuvre du respect de la confidentialité des données par les statisticiens. Il s'agit d'appliquer des procédures pour empêcher les personnes non habilitées d'identifier des informations individuelles à partir de résultats produits. Les méthodes suppressives, qui consistent à masquer l'information contenue dans des cellules de tableau, sont très majoritairement utilisées au sein du service statistique public français et décrites dans le **Guide du secret statistique** diffusé par l'Insee.

Dans un premier temps, le secret dit primaire doit s'appliquer conformément aux règles de fréquence minimale et de dominance. Dans le cas des statistiques d'entreprises, dans chaque cellule, il faut au minimum trois unités et que l'unité dominante ne représente pas plus de 85 % de la valeur diffusée, en tenant compte des éventuelles pondérations. L'identification des cellules concernées ne pose généralement pas de difficulté technique en termes de programmation.

En revanche, par la suite, le secret secondaire (ou induit) généré par la publication de sous-totaux ou par les niveaux hiérarchiques des nomenclatures utilisées est plus compliqué à traiter. Il faut en effet masquer plusieurs cellules supplémentaires pour ne pas retrouver par différence les valeurs des cellules sous secret primaire.
Dans le cadre de la démarche générale du ministère de l'Agriculture de transfert de programmes en R, une remise à plat du processus a été effectuée pour remplacer une macro SAS qui appliquait depuis plusieurs années les méthodes suppressives aux tableaux diffusés.
Des méthodes perturbatrices de type « cellule clé » ont également été envisagées et testées avec R, comme la Cell-key method (CKM), développée par l'Australian Bureau of Statistics. Elle a notamment été mise en avant par le groupe de travail européen sur la diffusion du recensement de la population 2020. Une mise en œuvre est disponible dans R via le package cellKey, généralisation de la méthode du service statistique australien. La méthode CKM attribue à chaque observation (niveau individuel) un nombre aléatoire, appelé clé d'enregistrement (record key), compris entre 1 et une valeur nommée max cell key (par exemple 200), avec une probabilité uniforme. Lors d'une tabulation, chaque cellule se voit affecter un nombre égal à la somme des clés d'enregistrement modulo la valeur de la max cell key (cette valeur sommée conserve une probabilité uniforme). En fonction du nombre d'enregistrements constituant la cellule tabulée et de la valeur de sa cell-key, un bruit est ajouté à la valeur de la cellule, selon une table de perturbation préalablement construite avec le package **ptable**. Les données tabulées peuvent correspondre à des fréquences ou des magnitudes.

L'expérimentation s'est heurtée à deux difficultés : d'une part une relative complexité de la méthode, si bien qu'il a paru difficilement envisageable que des statisticiens non experts puisse en avoir la pleine maîtrise, et aussi une difficulté à en expliquer aux usagers les principes, et d'autre part, l'absence de règles heuristiques simples pour définir un niveau de perturbation acceptable, à l'instar des méthodes suppressives.

Les traditionnelles méthodes suppressives ont été alors privilégiées malgré le paysage complexe du traitement algorithmique du secret induit.
En effet, l'objectif du secret induit est de masquer d'autres cellules mais de sorte que l'intervalle de protection (ensemble des valeurs que l'on peut déduire) soit suffisamment grand mais que la perte d'information (en nombre de cellules, d'individus …) soit limitée. La suppression des cellules s'apparente à un programme de minimisation d'une fonction (la perte d'information) sous contraintes (les cellules dont les valeurs ne doivent pas être trouvées, même approximativement). Ce problème est à ce jour considéré comme un problème NP-complet, ce qui se traduit concrètement par une diversité d'algorithmes dont la performance dépend du cas concret à traiter (notamment le nombre de contraintes, et la taille du ou des tableaux à diffuser) [1] [2] [3].

L'algorithme le plus simple est la méthode HYPERCUBE. Elle traite séquentiellement chaque cellule sous secret primaire pour choisir les cellules adjacentes à masquer de la manière la plus appropriée. Mais si cette méthode est rapide et compréhensible, elle ne traite pas les tableaux liés et génère une forte perte d'information.
Il est généralement préférable d'utiliser un solveur d'équations dans les méthodes de type « Optimal » , « Modular » (ou HITAS) qui intègrent des algorithmes de plus court chemin de type « **Branch and Cut** » [4]. Mais si certains algorithmes protègent de manière exacte, d'autres font intervenir des intervalles de protection afin de ne pas pouvoir approcher la valeur à masquer. La définition d'un intervalle de protection doit se faire selon le contenu des données publiées et du risque associé de divulgation. Dans un processus où la pose de masque est généralement faite en dehors des bureaux métier, ce paramétrage métier a été considéré comme trop contraignant. À ce jour, la protection est faite pour empêcher le recalcul de manière exacte des valeurs. Lorsque le service aura davantage avancé sur la mise en oeuvre, il restera possible d'améliorer les traitements pour mieux protéger les données les plus sensibles.

1.2. Des outils pour appliquer les méthodes

τ-Argus est le logiciel de référence pour la protection de données tabulées. Son atout principal est de pouvoir masquer de façon optimale les cases grâce à une interface homme-machine sans connaissance préalable en programmation. En pratique, son utilisation est plutôt fastidieuse car elle nécessite de mettre sous un format approprié (ascii) séparément les micro-données (valeurs de la table) et les métadonnées (noms et caractéristiques des variables), ainsi que les fichiers hiérarchiques (nomenclatures des variables de croisement).

Des utilitaires ont néanmoins été développés depuis plusieurs années pour faciliter la création de ces fichiers en SAS ou en R mais aussi pour lancer des traitements automatisés en masse (mode batch).

Les avantages de R pour la mise sous secret des fichiers sont multiples :

- travailler dans le même environnement de travail que pour la préparation des données à diffuser, ce qui est intéressant dans une chaîne de traitements,
- disposer de fonctions prédéfinies pour mettre en forme les fichiers de données et les fichiers hiérarchiques avant le traitement global,
- garder une trace des traitements réalisés et pouvoir adapter facilement les scripts selon les cas.

Ces éléments ont motivé le choix des packages R pour la gestion du secret statistique dans les cubes de diffusion *Agreste Données en Ligne* (ADEL) dont la chaîne de traitement évolue progressivement vers R. Ce type de produit de diffusion, qui constitue une spécificité du SSP, permet de publier des données

multidimensionnelles avec lesquelles l'internaute peut interagir. L'outil Saïku lui permet notamment de sélectionner des variables d'affichage en ligne ou colonne et d'appliquer des filtres sur les variables pour sélectionner un sous-champ (une région ou un département par exemple pour la variable géographique). Actuellement, environ 400 cubes ADEL sont mis à disposition du grand public. Le nombre de dimensions (jusqu'à 4 ou 5) et la taille de certains tableaux (plus de 600 000 cellules pour l'exemple du recensement agricole ci-dessous) présentent un défi à part entière pour la mise en œuvre du secret statistique.

2. Premiers pas dans l'utilisation de sdcTable

Le package **sdcTable** téléchargeable sur le CRAN[1] s'applique à des données individuelles pour lesquelles on désire masquer d'éventuelles cellules (en les tabulant selon plusieurs variables de croisement à l'instar de τ-Argus).

2.1. Définir les hiérarchies

Une fois identifiées les variables à représenter dans le (ou les) tableau(x) croisé(s), la première étape est de définir la liste des dimensions (ou variables de croisement) qui vont définir l'hypercube à secrétiser.
Le package *sdcHierarchies,* qui est une dépendance de sdcTable, permet de construire assez facilement ces listes.

Dans le cas d'une dimension simplement constituée des modalités d'une variable (var1) et de son total (cas d'une nomenclature à 1 niveau de profondeur), la liste dim1 sera initialisée de la manière suivante :

*dim1<-**hier_create**(root = "Total", nodes = levels(as.factor(tab$var1))*

Si une dimension est constituée de modalités d'une variable regroupées à différents niveaux pour former des sous-totaux, on a affaire à un fichier hiérarchique qui doit être créé selon des règles strictes.
Le cas le plus courant est celui de la dimension géographique avec des niveaux de diffusion emboîtés.
Une hiérarchie peut être créée à partir d'un code avec la fonction **hier_compute** et la position du dernier caractère :

- Position 1 : Métropole=0, DOM=1
- Position 3 : Code région
- Position 5 : Code département
- Position 8 : Code commune

04408320

La hiérarchie peut se représenter sous la forme d'un dataframe ou d'un objet que l'on peut afficher grâce à la fonction **hier_dipslay.**
La visualisation de cette liste peut aussi se faire par un dataframe dont le niveau hiérarchique est représenté par le pointeur @.

```
|    +-89484
|    \-89486
\-9
  +-900
  | +-90001
  | +-90002
  | +-90003
```

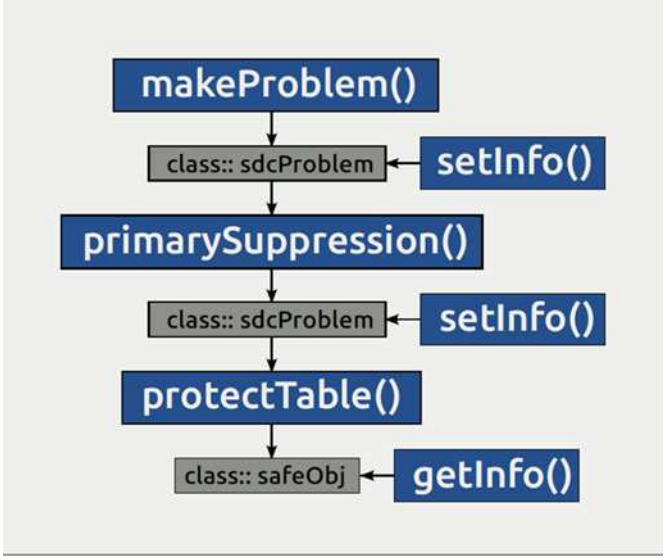
level	name
@	France entière
@@	0
@@@	011
@@@@	01177
@@@@@	01177001
@@@@@	01177002

On peut enfin dans certains cas supprimer des nœuds à une hiérarchie (**hier_delete**) ou en créer d'autres manuellement (**hier_create**) si les modalités ne sont pas présentes dans la variable.
Toutes les dimensions avec ou sans structure hiérarchique sont enfin stockées dans une liste utilisée par la suite pour définir les croisements.

[1] The Comprehensive R Archive Network https://cran.r-project.org/web/packages/sdcTable/index.html

2.2. Les différentes étapes d'application du secret

Le schéma ci-dessous présente les différentes étapes d'une utilisation classique du package **sdcTable**[2] :



- création de l'hypercube par la fonction **makeProblem()** en indiquant la table de données individuelles, la liste des dimensions (modalités avec ou sans hiérarchie), le nom des variables de la table associées aux dimensions et enfin la variable numérique. L'objet en sortie est de type ***sdcProblem.*** Il est difficilement manipulable mais certaines fonctions permettent d'en extraire les informations utiles sous un format lisible.

- application à l'hypercube des règles de secret primaire par la fonction **primarySuppression()**

type=«freq», maxN=2 #fréquence minimale de 2 unités incluses
type=«p», p=85 #dominance du plus gros contributeur à 85 %

Ces deux étapes reprennent les fonctionnalités de la fenêtre «*Specify tables*» dans τ-Argus dans laquelle on fait glisser les éléments des tableaux croisés et on spécifie les paramètres du secret primaire.

- application à l'objet des règles de secret secondaire par la fonction **protectTable()**

Les méthodes de τ-Argus sont disponibles (HYPERCUBE, OPT, HITAS) mais aussi un algorithme spécifique SIMPLEHEURISTIC proche de HITAS dans le fonctionnement, cependant peu documenté dans le package. Celui-ci est basé sur la méthode qui semble généralement la plus efficace car la plus rapide et la moins coûteuse en perte d'information. En revanche, il garantit uniquement que le recalcul exact des cellules sous secret primaire soit impossible. Il est toujours possible dans certains cas d'avoir une estimation approchée, alors que d'autres algorithmes permettent de définir des intervalles de protection autour des cellules sous secret. Pour limiter les difficultés à reconstruire le dispositif de pose de masque dans un cube multidimensionnel en R, l'algorithme simpleheuristic a été privilégié.

L'objet créé de type ***safeObj*** peut être récupéré par une fonction dans une table avec les valeurs tabulées, les effectifs et le statut de la case (s, u ou x).

- "u": cellule protégée (à masquer) en raison du secret primaire
- "x": cellule protégée (à masquer) en raison du secret secondaire
- "s": cellule qui n'est pas sous secret et qui peut être diffusée

[2] Vignette SdcTable https://cran.r-project.org/web/packages/sdcTable/vignettes/sdcTable.html

Ces étapes sont incontournables dans l'utilisation des méthodes mais on peut être confronté à des cas plus complexes.

3. Prise en compte de contraintes supplémentaires

3.1. La protection des cellules

Sans que cela ne soit imposé formellement par une quelconque règle, le producteur de données peut souhaiter ne pas masquer certaines cellules (comme des marges) qu'il considère plus importantes que les autres.
En d'autres termes, entre l'étape de secret primaire et de secret secondaire, il veut absolument pouvoir publier certaines cellules (comme par exemple des sous-totaux) même si ces dernières pourraient être masquées en raison du secret induit.
C'est une contrainte supplémentaire qui s'ajoute au programme d'optimisation au risque de supprimer davantage d'informations qu'initialement.
À ce titre, il y a attribution d'un statut particulier ('z') aux cellules à protéger dans l'objet en sortie de **primarySuppression.** La fonction ***change_cellstatus*** permet plus généralement de modifier le statut d'une ou plusieurs cellules identifiées par l'option ***specs.***
Dans un but de contrôle, la fonction ***cell_info*** donne le statut d'un groupe de cellules identifiées par ***specs.***

La protection des zéros est un cas particulier qui peut poser problème. En effet, hormis si la valeur nulle est sous secret primaire et ne concerne qu'une ou deux unités, les cellules avec une valeur nulle ne sont pas systématiquement masquées. Cette situation est gérée directement pour les deux secrets dans les fonctions ***primarySuppression*** *et* ***protectTable*** *avec l'option* ***protectZeros****=T.*

Un autre exemple concerne la diffusion de données par dérogation en levant le secret sur certaines modalités de variables à des échelons précis. Ainsi, on peut être amené à forcer le statut des cellules concernées pour les rendre diffusibles. Par exemple, par avis du Comité du secret statistique[3], à l'échelon communal sur le Recensement Agricole 2020, le secret ne s'applique plus, entre autres, sur les superficies agricoles en céréales et oléo-protéagineux, en cultures permanentes et en prairies.

3.2. Les tableaux liés

Le package sdcTable offre la possibilité de traiter simultanément le secret secondaire dans deux tableaux liés avec au moins une dimension commune.
La fonction **protectLinkedTables()** est utilisable en précisant les cellules présentes dans les deux tables (***commonCells***). En sortie, on obtient une liste avec les deux tables secrétisées.
Cette fonction n'a pas encore été utilisée au SSP pour pouvoir en faire un retour plus détaillé.

3.3. Le cas des données pondérées

Il peut arriver que les données individuelles que l'on s'apprête à manipuler soient issues d'une enquête par méthode des sondages ou d'une enquête exhaustive mais dans laquelle la non-réponse a été traitée par repondération. Dans ce cas, à chaque unité, est affecté un poids de sondage qui intervient dans le traitement du secret.
Dans la fonction **makeProblem**, on ajoute deux lignes de code dans la fonction pour indiquer la position du coefficient dans la table à la fois pour le calcul des valeurs et l'application du secret primaire (***sampWeightInd***) et pour estimer la perte d'information lors du secret secondaire (***weightInd***). Généralement, le même jeu de pondérations est utilisé dans les deux traitements.

Ces coefficients sont pris en compte dans la règle de fréquence minimale pour dénombrer les unités dans chaque cellule. La règle de dominance est en revanche plus compliquée à appliquer.

[3] Avis modifié du 14 juin 2021 sur la diffusion de données communales issues de l'exploitation du recensement de l'agriculture de 2020

En effet, il faut identifier le plus gros contributeur au total d'une cellule (ainsi que sa contribution) lorsque les unités sont pondérées.
Pour ce faire, le package **sdcTable** doit utiliser des poids entiers afin de classer chaque unité selon l'ordre décroissant de sa contribution. Si les poids ne sont pas entiers (ce qui est très souvent le cas), le package effectue actuellement un arrondi alternativement par excès et par défaut. L'application de ces poids arrondis pose alors problème car le critère de dominance s'applique de manière imprécise. Une méthode de contournement a été trouvée et présentée ci-après.

4. Application aux données agricoles

L'intérêt principal du package est d'automatiser plus facilement qu'auparavant l'identification des cellules à masquer dans les données tabulées mises en ligne, et d'utiliser une mise en oeuvre reconnu par Eurostat. L'hypercube construit par croisement des dimensions est alors préalablement secrétisé avant d'être intégré dans l'outil de diffusion des tableaux dynamiques, ADEL.
Les premiers tests ont été réalisés sur l'enquête annuelle de branche EXF/SRI relative aux exploitations forestières et scieries.

4.1. Le cube relatif aux sciages

Le package sdcTable a été appliqué dans un premier temps sur la production de sciages de bois (variable ESSLIV) des scieries par essence et région en 2019.
Dans ce fichier, le code relatif à l'essence de bois (sur quatre caractères) peut être hiérarchisé selon les sous-totaux diffusés (feuillus tempérés > chêne > plots, avivés, autres …) par une chaîne de caractères dont la longueur indique le niveau hiérarchique (1, 4, 6 et 9).

Nomenclature	Hierarchie	ESSENCE
Sciages	1	3700
Sciages de feuillus tempérés	1.11	2350
Sciages de chêne	1.11.1	2170
Plots de chêne	1.11.1.11	2120
Avivés de chêne	1.11.1.12	2150
Autres sciages de chêne	1.11.1.19	2160
Sciages de hêtre	1.11.2	2250

La fonction **hier_compute** avec l'option **endpos=** permet de construire aisément la dimension hiérarchisée.

```
dim.prod<- hier_compute(inp = levels(as.factor(sci19$Hierarchie)),
                        dim_spec = c(1,4,6,9),root = "Total",method = "endpos")
```

La dimension géographique se réduit aux modalités du code région REG et au total France entière.

```
dim.reg<-hier_create(root = "France entière", nodes = levels(as.factor(sci19$REG)))
```

L'hypercube doit préciser les dimensions créées et les associer aux variables du fichier

*p <- **makeProblem**(data =sci19, dimList = list(REG=dim.reg, Hierarchie= dim.prod), dimVarInd=c(which(colnames(sci19)=="REG"),which(colnames(sci19)=="Hierarchie")), numVarInd = "ESSLIV")*

Les règles habituelles de secret primaire sont ensuite appliquées à l'objet p pour former l'objet problem_supp traité pour le secret secondaire par la méthode SIMPLEHEURISTIC.

En sortie, le fichier précisant le statut de diffusion de chaque cellule peut être récupéré pour la suite des traitements.

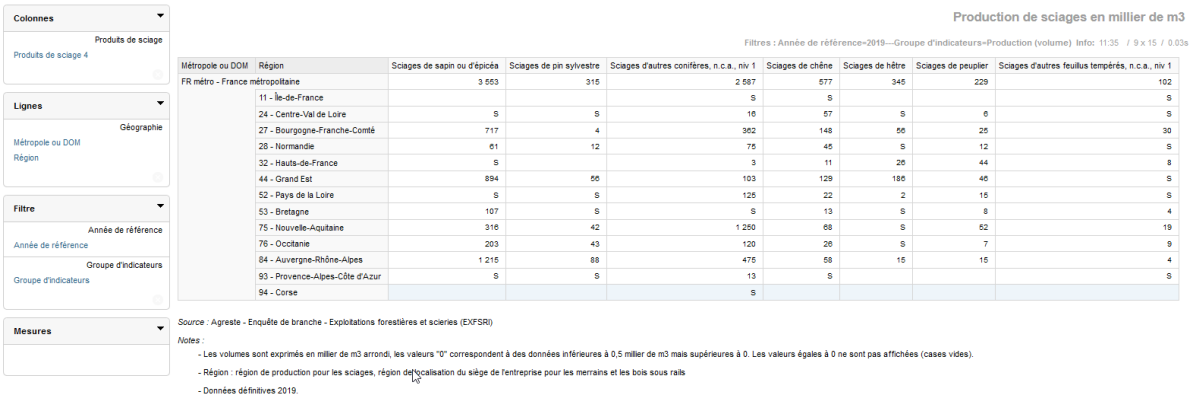
REG	Hierarchie	Freq	ESSLIV	sdcStatus
France entière	Total	26775	7 707 760	s
11	Total	63	367	x
11	1.11	33	266	u
11	1.11.1	9	251	u
11	1.11.1.11	3	22	u
11	1.11.1.12	3	0	s

Le bilan des suppressions peut s'effectuer en dénombrant le nombre de cellules n et le total de la variable tot selon le statut.

sdcStatus	n	tot
s	248	60 115 540
u	125	1 625 317
x	30	1 327 896

Sur le tableau des sciages par région et essence en 2019, 248 cellules peuvent être diffusées, 125 doivent être masquées en raison du secret primaire et 30 supplémentaires par secret induit.

Après intégration dans l'outil, voici un exemple à l'écran de tableau dynamique



Mais, quelques mois plus tard, le traitement du millésime 2020 a posé une difficulté qui n'avait pas été anticipée en raison des pondérations associées à la variable relative aux volumes.

En effet, pour l'enquête 2020, le traitement de la réponse qui se faisait auparavant par imputation s'est fait par repondération (à partir de poids initiaux égaux à 1) par la méthode des Groupes de réponse homogène.

L'intégration de coefficients de pondération est techniquement possible dans l'hypercube et le traitement du secret par sdcTable, mais des problèmes ont été constatés.

Tout d'abord, la règle de dominance ne masquait pas toujours les mêmes cellules, ce qui était plutôt surprenant. Après recherche, il semblait que les coefficients devaient être arrondis car l'algorithme nécessitait des valeurs entières pour classer les contributions. Cet arrondi était effectué aléatoirement par excès ou par défaut. Ce « bug » a été corrigé dans la dernière version 0.32.2 du package de décembre 2021 en arrondissant cette fois de manière déterministe (voir paragraphe 3.3).

Cependant le nombre de cellules masquées par la règle de dominance semblait encore trop élevé, ce qui fut confirmé par comparaison avec le même traitement réalisé avec τ-Argus.
Du fait des arrondis, on observait que la règle de dominance devenait très approximative avec des coefficients de pondération faibles.

Une solution de contournement a cependant été trouvée en rendant ces pondérations entières. Il suffit d'opérer ainsi. Les coefficients avec trois décimales sont multipliés par 1 000 et la variable numérique divisée par cette même valeur 1 000.

*sci20$COEF=sci20$COEF*1000*
sci20$ESSLIV=sci20$ESSLIV/1000

De cette manière, il n'y a plus de problème lié aux arrondis et les valeurs des cases, sommes pondérées, sont identiques.

Cela revient à modifier les règles de secret primaire sur des données dont les pondérations ont été multipliées par 1000 :

- **La règle de fréquence minimale à 3 unités devient une règle de fréquence minimale à 3 000 unités**[4]

primarySuppression(problem_supp,type = 'freq',maxN = 3000)

- **La règle de dominance à 85 % dite (1,85) devient une règle de dominance (1000,85) dans laquelle les 1000 premiers contributeurs ne doivent pas dépasser 85 % de la valeur totale**

primarySuppression(problem_supp,type='nk',n=1000,k=85,numVarName="ESSLIV")

Au total, sur 400 cellules à diffuser, 47 sont masquées par secret primaire et 31 autres par secret induit.

4.2. Le cube du Recensement Agricole

Les premiers résultats du Recensement Agricole 2020 ont aussi nécessité d'actualiser les tableaux diffusés sur le site Agreste.

Dans ce cube, les variables de croisement sont :

- les niveaux géographiques (voir paragraphe 2.1),
  France entière > Métropole ou DOM > Région > Département > Commune
- l'orientation technico-économique (OTEX) dominante (grandes cultures, viticulture …),
- les classes de taille économique : micro, petites, moyennes et grandes exploitations.

Les indicateurs représentés sont (avec le nombre d'exploitations) en 2010 et 2020 :

- Superficie agricole utilisée (SAU) (en hectare),
- Équivalent temps plein (ETP),
- Production brute standard (PBS) (en millier d'euros).

En pratique, le traitement a nécessité la construction d'un hypercube pour chaque indicateur car le statut d'une cellule peut différer selon la variable.
De plus, en raison des multiples niveaux hiérarchiques de la géographie ainsi que du grand nombre de modalités, l'application de la règle de dominance est gourmande en mémoire vive et prend plusieurs heures de traitement. Malgré ces contraintes, la mise sous secret des données du Recensement agricole ne nous a pas, jusqu'ici, posé de problèmes particuliers.

[4] **Il faut prendre le seuil de 3 000 pour considérer tous les poids compris entre 2 inclus et 3 exclu.**

5. Conclusions et perspectives

En conclusion, les différents travaux réalisés se sont révélés plutôt concluants et offrent des perspectives dans la secrétisation future d'autres fichiers de données.

La macro SAS utilisée jusqu'à présent pour traiter le secret était si complexe que le service ne parvenait plus à identifier les différentes fonctionnalités et à s'assurer que la protection était correctement faite. L'automatisation faite avec R est un avantage non négligeable en temps de traitement et de répartition des travaux au sein du service, puisque la mise sous secret peut être actualisée à chaque nouveau millésime par les responsables d'enquête eux-mêmes, sans faire intervenir d'expert sur le traitement du secret statistique. Elle permet alors une application du secret statistique plus efficace et plus sûre qu'avec les procédures jusqu'alors en vigueur, d'autant que le package sdcTable fait partie des outils recommandés par Eurostat[5],

La mise en ligne d'autres cubes du Recensement agricole est prévue dans les prochains mois.
De même pour l'enquête annuelle EXF/SRI, les millésimes de 2005 à 2020 seront progressivement mis en ligne avec les nouvelles méthodes de protection.

Bibliographie

[1] *CSO Best Practice for Statistical Disclosure Control of Tabular Data* - Timothy Linehan and Karina Dineen : Office statistique irlandais

[2] *Privacy in Statistical Databases. CASC Project International Workshop*, PSD 2004, Barcelona, Spain, June 9-11, 2004, Proceedings by Josep Domingo-ferrer, Vicenc Torra

[3] *CSO Best Practice for Statistical Disclosure Control of Tabular Data*
Timothy Linehan and Karina Dineen

[4] A Branch-and-Cut Algorithm for the Symmetric Generalized Traveling Salesman Problem, Matteo Fischetti, Juan José Salazar González et Paolo Toth, in Operations Research, 1997, vol. 45, issue 3, 378-394

Quelques liens sur le package sdcTable
https://github.com/cran/sdcTable
https://cran.r-project.org/web/packages/sdcTable/vignettes/sdcTable.html
https://www.r-pkg.org/pkg/sdcTable
https://unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/2011/10_Meindl.pdf

[5] https://joinup.ec.europa.eu/collection/statistics/solution/sdctools-tools-statistical-disclosure-control/about