Macro Tau_Argus : Cas pratiques

Le présent document a pour objet d'illustrer l'utilisation de la macro sas *%tau_argus*. Nous présenterons des cas simples et courants, ainsi que des exemples d'utilisation de la macro dans des cas particuliers et plus complexes.

Si certains exemples ne vous paraissent pas suffisamment clairs, ou que vous souhaiteriez voir d'autres exemples spécifiques, n'hésitez pas à nous faire un retour par mail (-dg75-l120@insee.fr).

La macro *%tau_argus*, ainsi que la table sas qui servira de base pour les exemples présentés ici, sont téléchargeables sur l'intranet de la division RTI du DMS (adresse valable en octobre 2017) :

https://www.agora.insee.fr/cms/sites/dmcsi/home/DMS/DRTI/ConfidentialiteDonnees.html

Les différents paramètres de la macro sont explicités en début de programme, ainsi que quelques conseils et remarques en fin de programme. Pour savoir comment écrire les paramètres, bien se référer aux annotations en début de programme.

Après chaque appel de la macro, toujours regarder la log sas. Le programme est relativement bien fourni en messages explicites quant aux éventuels problèmes ou erreurs.

Pour les illustrations présentées ici, nous travaillerons sous AUS dans le répertoire :

U:\tests macro pour intranet

S'y trouveront donc, la table sas de données individuelles, ainsi que les différents fichiers et répertoires créés par la macro.

Nous nous servons ici de la version 3.5 de Tau-Argus, parce que cela correspond à nos habitudes de travail. Cependant, il est possible d'utiliser des versions plus récentes du logiciel, ce qui a notamment pour intérêt l'accès gratuit aux solveurs complexes *modular* et *optimal*, uniquement disponibles via une clef payante (XPRESS) dans la version 3.5. Ces solveurs donnent de meilleurs résultats en termes de minimisation de la perte d'information pour le secret secondaire, mais sont plus lents à faire tourner.

Pour se faire, il est nécessaire de télécharger la dernière version de Tau-Argus "with bundled JRE7" (pour des raisons de compatibilité Java), sur le site de CBS http://neon.vb.cbs.nl/casc/tau.htm, et de la dézipper dans un répertoire accessible sous AUS ("U:\tau-argus 4.1.6" par exemple). Il suffira, d'ajouter les paramètres suivants aux différents appels de macro :

```
TauArgus_exe            =   U:\tau-argus 4.1.6\TauArgus.exe,
TauArgus_version        =   opensource,
```

Par défaut, le solveur utilisé pour le secret secondaire sera *hypercube*. Pour utiliser *modular* (dans le cas de tableaux liés notamment) ou *optimal*, il suffira d'ajouter le paramètre :

```
solver                  =   modular /*ou optimal*/,
```

Cependant, il n'est pas exclu que le solveur ne fonctionne pas, parfois pour des raisons de configuration de tableau, de nombre de variable de ventilation, de nombre de case ... pas

de recette magique ici, il faudra tester les solveurs un par un pour choisir celui qui donnera les meilleurs résultats en termes d'optimisation du secret secondaire et de lourdeur du processus.

Présentation de la table *legumes.sas7bdat*

Pour les besoins d'illustrations, nous mettons à disposition une table de données individuelles, entièrement fictive. L'individu statistique est ici une entreprise qui vend des légumes. Cette table n'a pour unique objectif que de fournir un support pour illustrer la mise en oeuvre de la macro sas, les données n'ont aucune vraisemblance.

Les variables catégorielles / variables de ventilation :

- *A10*, *A21*, *A38*, *A88*, *A129*, *A272*, *A615* et *A732* : secteur d'activité à différents niveaux d'agrégation (parce que oui, une entreprise du bâtiment peut aussi vendre des *TOMATES*...), du plus agrégé au plus détaillé.

- *TYPE_DISTRIB* : l'histoire ne dit pas à quoi correspondent les codes...

- *TREFF* : tranche d'effectifs.

- *CJ* : catégorie juridique.

- *NUTS0*, *NUTS1*, *NUTS2* et *NUTS3* : nomenclature des unités territoriales statistiques, du plus agrégé au plus détaillé.

- *PAYS* : pays...

- *DEP* : département...

Les variables de réponses / variables numériques.

- *MACHES*, *BATAVIAS*, *TOMATES*, *POIVRONS* et *RADIS* : les légumes...

- *LEGUMES_ROUGES* : somme de *TOMATES*, *POIVRONS* et *RADIS*.

- *SALADES* : somme de *MACHES* et *BATAVIAS*.

- *LEGUMES_TOTAL* : somme de *LEGUMES_ROUGES* et *SALADES*.

- *PIZZAS* : histoire de se diversifier...

Les autres variables pouvant être utiles à Tau-Argus

- *IDENT* : identifiant (il s'agit en fait du numéro de ligne). Notez qu'il est important que la table soit triée par l'identifiant si l'on souhaite se servir de l'option "holding".

- *POIDS* : poids (au sens sondage i.e pondération)...

Chargement de la macro sous sas

Avant toute utilisation de la macro, il est nécessaire de charger la macro sas. Soit en l'exécutant simplement, soit en exécutant les lignes suivantes :

```
option mprint;
filename tauargus "U:\tests macro pour intranet"; /* répertoire où se trouve la macro */
%include tauargus (Macro_Tau_Argus);
```

Le cas simple : appliquer le secret sur un tableau simple

On souhaite diffuser un tableau ventilant le nombre de *TOMATES* par secteur d'activité regroupé en *A21* et par *PAYS*.

```
%TAU_ARGUS (
tabsas                =       legumes,
library               =       U:\tests macro pour intranet,
tabulation_1          =       a21 pays tomates);
```

Le cas simple : appliquer le secret sur un tableau simple

On souhaite diffuser un tableau ventilant le nombre de *TOMATES* par secteur d'activité regroupé en *A21* et par *PAYS*... mais on souhaite lui appliquer des règles de secret primaires particulières :

- une dominance à 80 % maximum,

- au moins 11 individus par case,

- une règle du P% à 20 % (le deuxième contributeur de la case ne doit pas être capable d'estimer la valeur du premier à moins de 20 % près).

```
%TAU_ARGUS (
tabsas                    =   legumes,
library                   =   U:\tests macro pour intranet,
tabulation_1              =   a21 pays tomates,
primary_secret_rules      =   DOM P FREQ,
dom_k                     =   80,
p_p                       =   20,
frequency                 =   11);
```

D'autres paramètres sont disponibles pour ajuster les règles et sont bien explicités en début de programme. Les règles sont par défaut celles de la statistique d'entreprise (règle de fréquence : pas moins de trois entreprises par case ; règle de dominance : le premier contributeur de la case qui ne doit pas faire plus de 85 % du total de la case).

Le cas simple : appliquer le secret sur un tableau de comptage

On souhaite diffuser un tableau ventilant le nombre d'entreprise par secteur d'activité regroupé en *A21* et par *PAYS*. Il n'y a pas dans la table sas de variable correspondant au "nombre d'entreprise", puisque chaque ligne vaut pour 1. Il suffit alors de compléter la ventilation par "freq". La règle de secret primaire de dominance ne sera alors pas appliquée, puisque non pertinente.

```
%TAU_ARGUS (
tabsas                =       legumes,
library               =       U:\tests macro pour intranet,
tabulation_1          =       a21 pays freq);
```

Le cas simple : appliquer le secret sur deux tableaux liés

On souhaite diffuser deux tableaux liés par leur marges :

- un ventilant le nombre de *TOMATES* par secteur d'activité regroupés en *A21* et par *PAYS,*

- un ventilant le nombre de *TOMATES* par secteur d'activité regroupés en *A21* et par catégorie juridique.

```
%TAU_ARGUS (
tabsas              =       legumes,
library             =       U:\tests macro pour intranet,
tabulation_1        =       a21 pays tomates,
tabulation_2        =       a21 CJ tomates);
```

Par défaut, si les variables de réponse (ici *TOMATES*) sont les mêmes pour l'ensemble des tabulations, Tau-Argus traitera le secret secondaire en mode "tableaux liés", c'est à dire que le logiciel va reconstruire le tableau regroupant toutes les ventilations concernées (ici un tableau ventilant *A21*, *PAYS* et *CJ*) pour ensuite poser le secret secondaire.

La limite du nombre de ventilations est de six avec l'algorithme *hypercube*, et de quatre avec les algorithmes *modular* et *optimal*.

Le cas simple : appliquer le secret sur un tableau ventilant une variable "hiérarchique"

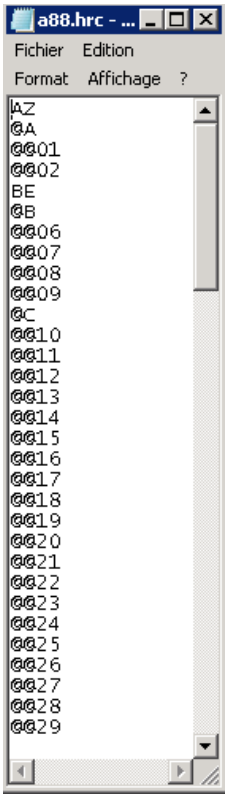
On souhaite diffuser le nombre de *TOMATES* par secteur d'activité à différents niveaux d'agrégation : *A10*, *A21*, *A88*.

```
%TAU_ARGUS (
tabsas              =       legumes,
library             =       U:\tests macro pour intranet,
tabulation_1        =       a88 tomates,
hierarchy_1         =       a10 a21 a88,
hierarchical_var    =       a88);
```

Pour Tau-Argus, il n'y a qu'une seule variable : le secteur d'activité. Les différents niveaux d'agrégations sont alors une propriété de cette variable et doivent être décrits dans un fichier plat, d'extension ".hrc".

Tau-Argus n'offre pas de solution pour générer ce type de fichier, on doit le faire "à la main". La macro %tau_argus contient des paramètres (*HIERARCHY_1* à *HIERARCHY_6*) permettant de générer des fichiers au bon format, à partir des variables des différents niveaux agrégés souhaités. Veillez à n'utiliser ce paramètre que dans le cas de hiérarchie simple et symétrique. Dans le cas de hiérarchie plus complexes (par exemple correspondant à une diffusion fine sur le secteur industrielle et agrégée sur les autres secteur), la macro ne permet pas de générer automatiquement le fichier plat de hiérarchie.

Extrait du fichier de hiérarchie créé :



Si l'on dispose déjà d'un fichier de hiérarchie, il n'est pas nécessaire d'en générer un. Le fichier devra s'intituler du nom de la variable la plus détaillée (ici : *A88*) avec l'extension ".hrc".

L'appel de la macro sera alors :

```
%TAU_ARGUS (
tabsas              =       legumes,
library             =       U:\tests macro pour intranet,
tabulation_1        =       a88 tomates,
hierarchical_var    =       a88);
```

Le cas complexe : appliquer le secret à un tableau contenant des cases négatives

On souhaite diffuser deux tableaux liés ventilant le nombre de *PIZZAS*

- par *NUTS3* et *TYPE_DISTRIB*

- par A*88* et *NUTS*

Le problème, c'est que certaines entreprises ont réussi (par on ne sait quel miracle) à produire moins de *PIZZAS* que 0 (c'est juste pour l'exemple)... On se retrouve alors avec des cases négatives, et lors de l'application secret secondaire, notamment lorsqu'il s'agit de tableaux liés, Tau-Argus n'arrive pas à trouver de solution.

Nous avons mis en place, au DMS, une solution complexe mais permettant de contourner le problème. Cette solution est présentée sous la forme d'une macro sas (*%tau_argus_negatives*) téléchargeable sur la page de l'intranet.

On charge donc cette macro.

```
option mprint;
filename tauneg "U:\tests macro pour intranet"; /* répertoire où se trouve la macro */
%include tauneg (Macro_Tau_Argus_negatives);
```

... et on l'appelle. Les paramètres disponibles sont spécifiés en début de programme. L'appel de la macro *%tau_argus* est moins flexible, de nombreux paramètres sont inscrits en dur dans la macro *%tau_argus_negatives*.

```
%TAU_ARGUS_NEGATIVES (
library             =       U:\tests macro pour intranet,
tabsas              =       legumes,
tabulation_1        =       nuts3 type_distrib pizzas,
tabulation_2        =       a88 nuts0 pizzas) ;
```

La méthode consiste à faire le secret primaire en double : sur la variable de réponse telle quelle d'une part, et sur la variable de réponse en valeur absolue d'autre part. On compile ensuite les résultats avec comme règle : si la case est cachée dans au moins un des deux cas de figure, alors on cache. Pour le secret secondaire, nous calculons une variable de coût de secret secondaire à partir de la variable de réponse problématique. Les valeurs des différents individus sont modifiées de sorte qu'il n'y ait plus de valeur négatives, que l'ordre des individus soit respecté et que le grand total soit identique, selon la formule suivante : Vi'=[(Vi-min(Vi)) x somme(Vi)] / [somme(Vi-min(Vi))] (Vi étant la valeur pour un individu i. Les masques de secret produit présentent néanmoins la variable de réponse initiale pour faciliter les éventuels contrôles.

Le cas complexe : appliquer le secret à des tableaux liés par une variable mais pas au même niveau d'agrégation

On souhaite diffuser les deux tableaux liés par leur marge :

- un ventilant le nombre de *RADIS* par *NUTS3* et par type de distribution

- un ventilant le nombre de *RADIS* par *NUTS2* et par *A88.*

Le problème est que si on fait l'appel simple suivant :

```
%TAU_ARGUS (
library             =       U:\tests macro pour intranet,
tabsas              =       legumes,
tabulation_1        =       nuts3 type_distrib radis,
tabulation_2        =       nuts2 a88 radis) ;
```

Tau-Argus considèrera les deux variables *NUTS2* et *NUTS3* comme indépendantes, et de possibles problèmes d'incohérence, notamment à cause du secret secondaire, pourraient se produire.

Il est possible en mode "clique bouton" de gérer ce problème de diffusion à des niveaux agrégés différents d'un tableau à l'autre en passant par l'option de recodage que propose Tau-Argus (cf. Le tutoriel sur Tau-Argus, ou encore les support de formation téléchargeables sur l'intranet). Cependant, cette option n'est pas disponible dans le mode "batch" du logiciel (mode dont on se sert dans la macro SAS).

Pour contourner ce problème, on peut utiliser la macro en deux temps, en passant par le mode "tabledata", c'est-à-dire que l'on présentera des données déjà tabulées en entrée de

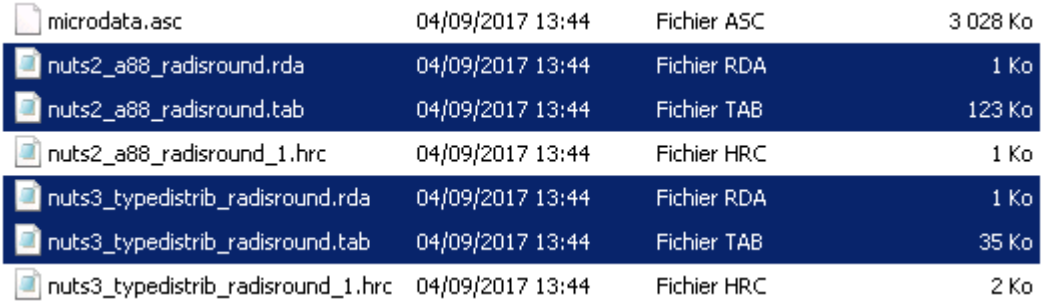
Tau-Argus. Ce mode présente certains avantages, mais est plutôt très sensible à l'exacte additivité des cases internes du tableaux par rapport à ses marges. La variable *RADIS* ayant de nombreuses décimales, nous préfèrerons travailler ici sur la version arrondie de cette variable : *RADIS_ROUND*.

On fait un premier appel pour générer des fichiers plats de données tabulées. C'est le paramètre *OUTPUTTYPE = 5* qui permet cela.

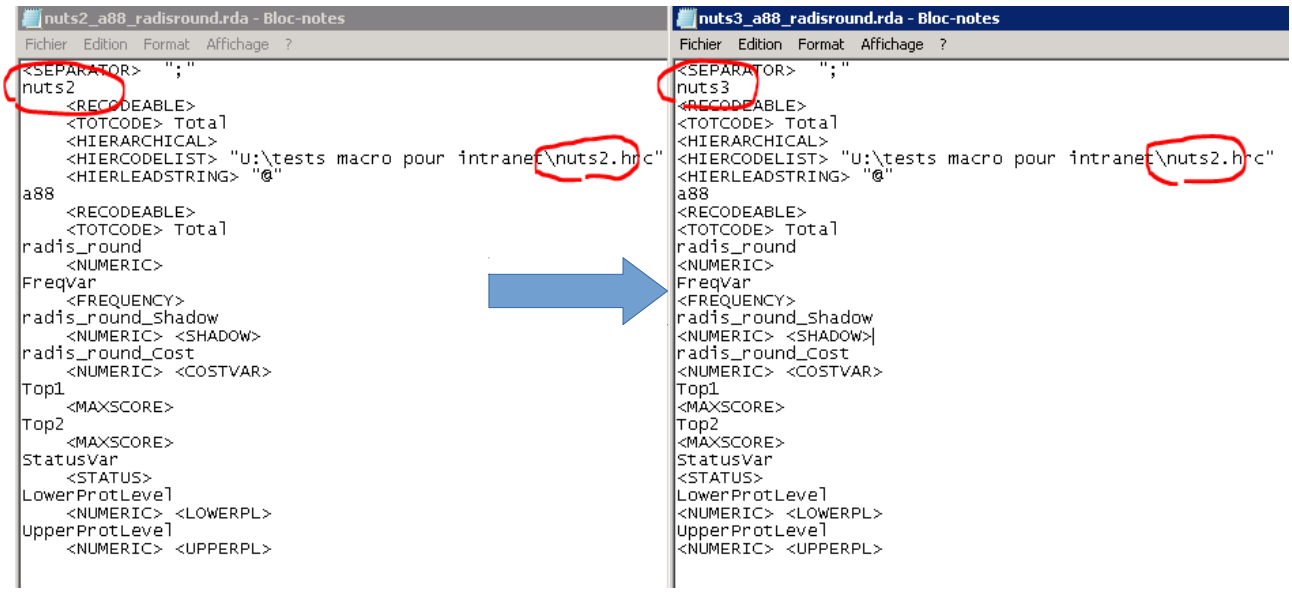
```
%TAU_ARGUS (
library             =       U:\tests macro pour intranet,
tabsas              =       legumes2,
tabulation_1        =       nuts3 type_distrib radis_round,
tabulation_2        =       nuts2 a88 radis_round,
hierarchical_var    =       nuts2 nuts3,
hierarchy_1         =       nuts0 nuts1 nuts2 nuts3,
hierarchy_2         =       nuts0 nuts1 nuts2 ,
solver              =       ,
outputtype          =       5) ;
```

On ne souhaite pas ici avoir le secret secondaire, on renseigne donc *SOLVER* = (vide). Pour notre problème, il est nécessaire d'avoir des hiérarchies distinctes dans un premier temps pour *NUTS2* et *NUTS3*.

Les deux sorties correspondant aux données tabulées, sont en fait, pour chaque tabulation, deux fichiers au bon format pour constituer un fichier d'entrée pour Tau-Argus : la tabulation (".tab") et ses métadonnées (".rda").



Ensuite, on fait une modification "manuelle" du fichier de métadonnées de la tabulation 2 : *NUTS2 A88 RADIS_ROUND.* Le nom de la variable devient *NUTS3*, pour que Tau-Argus considère qu'il s'agit bien de la même variable que dans l'autre tabulation (*NUTS2 TYPE_DISTRIB RADIS_ROUND*). En revanche on ne modifie pas le fichier de hiérarchie rattaché à cette tabulation... et c'est cette subtilité qui n'est pas possible lorsque l'on passe par le mode "microdata". Avec ce dernier, les différentes tabulations ont un seul et même fichier de métadonnées, ce qui ne permet pas d'affecter deux hiérarchies différentes à une même variable.



On peut faire ces changements à la main ou l'on peut également les programmer. Pour les besoins de la macro *%tau_argus*, les fichiers correspondant à la tabulation 2 devront s'intituler *NUTS2_A88_ RADISROUND* (".tab" et ".rda").

```
%macro change_rda (library, tabulation,vardep, vararr);
    /* cette étape data permet de convertir la tabulation (ici "nuts2 a88 radis") en une version qui correspond au
    nom de la table sas associé, sans espace entre les variables ("nuts2_a88_radis").*/
    data _null_ ;
        call symput ("output_name",compress(tranwrd(tranwrd(trim(tranwrd("&tabulation","_","**"))," ","_"),"**",""))) ;
    run ;
    proc import datafile    =   "&library.\TEMPORARY FILES MACRO\&output_name..rda"
                out         =   rda
                dbms        =   dlm replace;
                delimiter   =   '**';
                getnames    =   no;
    RUN;

    data rda ;
        set rda ;
        if var1 =   "&vardep" then var1 =   "&vararr";
    run;

    data _null_ ;
        call symput ("output_name2",tranwrd("&output_name","&vardep","&vararr")) ;
    run ;

    data _null_ ;
        File "&library.\TEMPORARY FILES MACRO\&output_name2..rda" dlm="" lrecl=200;
        set rda;
        Put (_all_)(+0);
    run;

    option noxwait xsync;
    X copy "&library.\TEMPORARY FILES MACRO\&output_name..tab"  "&library.\TEMPORARY FILES MACRO\&output_name2..tab";
%mend;
%change_rda (
library     =   U:\tests macro pour intranet,
tabulation  =   nuts2 a88 radis_round,
vardep      =   nuts2,
vararr      =   nuts3);
```

Enfin, on relance la macro %tau_argus, en spécifiant que l'entrée est une tabulation (*INPUT = tabledata*). Pas besoin ici de spécifier que telle ou telle variable est une variable de poids, de holding, hiérarchique... ces informations sont déjà contenues dans les fichiers de métadonnées (".rda") qui accompagnent les tabulations.

```
%TAU_ARGUS (
library                 =       U:\tests macro pour intranet,
input                   =       tabledata,
tabulation_1            =       nuts3 type_distrib radis_round,
tabulation_2            =       nuts3 a88 radis_round) ;
```

Le secret secondaire a donc bien été fait en tableaux liés sur ces deux tabulations, et les deux variables *NUTS2* et *NUTS3* sont bien considérées comme une même variable.

Le cas complexe : des tabulations liées non par les variables de ventilations mais par les variables de réponse.

On souhaite diffuser les trois tableaux suivant :

- on ventile les légumes rouges par *A88* et par *CJ*.

- on ventile les *SALADES* par *A88* et par *CJ*.

- on ventile les légumes totaux par *A88* et par *CJ*.

S'il y a bien un lien évident entre ces tableaux (*LEGUME_TOTAL = LEGUME_ROUGE + SALADE*), il n'existe pas de solution simple dans Tau-Argus pour gérer ce lien.

On peut cependant gérer la confidentialité de deux manières, l'une simple et peu coûteuse, l'autre plus complexe et réclamant plus de programmation.

La première consiste à ne faire le secret que sur l'un des trois tableaux et à appliquer le masque de secret ainsi obtenu aux trois autres. On pourrait également faire les trois secrets primaires, les compiler (si une case est cachée dans l'un des trois tableaux alors elle est cachée), et faire le secret secondaire avec une seule des trois variables de réponse comme variable de coût.

La seconde solution consiste à transformer les tableaux pour que Tau-Argus soit en capacité de gérer le lien entre les variables... On va ajouter une variable de ventilation qui renseignera sur le type de légumes et ne faire qu'une seule tabulation.

```
libname legumes "U:\tests macro pour intranet";

data legumes.legumes2 ;
    set legumes.legumes (in =   leg_r   rename  =   (legumes_rouges =   quantite))
        legumes.legumes (in =   salad   rename  =   (salades    =   quantite));
    if leg_r    =   1 then type_leg =   "legumes_rouges";
    if salad    =   1 then type_leg =   "salades";
run;

proc sort   data    =   legumes.legumes2 ; by ident ; run;

%TAU_ARGUS (
tabsas              =   legumes2,
library             =   U:\tests macro pour intranet,
tabulation_1        =   a88 cj type_leg quantite,
holding_var         =   ident );
```

Lors de l'appel de la macro, on ajoute le paramètre *HOLDING_VAR* puisque l'individu statistique de la nouvelle table legumes2 n'est plus l'entreprise, mais les ventes des entreprises par type de légumes. Ce paramètre permet alors à Tau-Argus de ne pas compter en double une entreprise qui serait présente dans une case à via deux types de légumes différents (ici, seule la marge est concernée par cet aspect).