Icarus : un package R pour le calage sur marges et ses variantes

Antoine Rebecq [1]

[1] *INSEE, 18, bd Adolphe Pinard. 75014 Paris. antoine.rebecq@insee.fr*

Résumé. On décrit les fonctionnalités du package R Icarus, qui implémente le calage sur marges ainsi que certaines de ses variantes : calage pénalisé, calage sur bornes serrées. Icarus est pensé comme un package facilitant le calage en production statistique. Son interface est proche de celle de la macro SAS Calmar. Afin d'aider le statisticien d'enquête à choisir des paramètres de calage, le package affiche des statistiques et graphiques liés à la qualité des estimateurs par calage. Le package étant destiné à la production de statistiques officielles, il est important de minimiser le risque d'une erreur de programmation. De nombreux tests unitaires ont donc été implémentés.

Mots-clés. R, Calage, Calmar, Estimation

1 Introduction

Le calage sur marges, introduit par Deville et Särndal en 1992 ([3]) est depuis largement utilisé en production en statistique publique et par les instituts de sondage. L'utilisation du calage en statistique publique permet de présenter des estimations cohérentes entre différentes sources, mais aussi d'augmenter la précision des estimations si le calage est effectué sur des variables bien liées aux variables d'enquête. Les macros Calmar [1] (1993, [12]) et Calmar2 (2003, [11]) implémentent le calage sur marges en SAS. Ces macros ont été créées avec la vocation d'être largement utilisées dans le système statistique public.

Le langage de programmation R possède beaucoup d'avantages et est de plus en plus utilisé en statistique publique. R est un logiciel libre et open source qui comporte un langage de programmation orienté vers la manipulation de données vectorielles. Contrairement à d'autres langages utilisés fréquemment en statistique, R permet la programmation fonctionnelle (et également orientée objet), ce qui facilite le développement de méthodes réutilisables et modulaires. Ces propriétés sont largement mises à profit par la communauté statistique. Ainsi, de nombreux packages implémentent les méthodes des diverses branches de la statistique, parfois à la pointe de la recherche. Par ailleurs, en production comme pour l'utilisateur moins avancé, R peut également être utilisé sous forme de scripts.

1. Disponible sur le site de l'INSEE : `http://www.insee.fr/fr/methodes/default.asp?page=outils/calmar/accueil_calmar.htm`

Concernant le calage sur marges, les packages R *sampling* ([**tille2015package**]) et *survey* ([8]) en proposent une implémentation. Toutefois, ces fonctions ne présentent pas d'interface spécifiquement pensée pour l'utilisateur non expert. Ainsi, il faut transformer et réorganiser les variables (particulièrement les variables catégorielles), contrairement à l'esprit dans lequel avait été pensé la macro Calmar. Le package Icarus [2] a été développé avec la même ambition que la macro SAS. Le mécanisme d'entrée des marges est ainsi très similaire entre Icarus et Calmar : il s'agit d'indiquer dans une matrice les totaux (ou les pourcentages) associés aux modalités de chaque variable (voir partie 2.2). Dans le cas où l'utilisateur entrerait plusieurs contraintes de marge colinéaires (redondantes), le package Icarus comme la Macro Calmar2 traitent le problème de façon transparente en utilisant l'inverse généralisée dans le calcul de la solution.

Le package Icarus est disponible sur le CRAN, en version 0.2.1 au 29 août 2016 [3]. Il peut être donc installé en utilisant la commande :

```
R > install.packages("icarus")
```

2 Calage sur marges

2.1 Notations

Considérons une population $\mathcal{U}$ de taille N pour laquelle on essaie de mesurer des caractères $Y \in \mathbb{R}^N$. Pour cela on tire un échantillon s de taille n avec un plan p de probabilités d'inclusion simples π_k. On se donne J variables auxiliaires (X_j) (définies sur toute la population, donc en particulier sur l'échantillon), desquelles on connaît les totaux $T(X_j)$. Le vecteur $T_X \in \mathbb{R}^J$ regroupe ces totaux (ou marges de calage). G désigne une pseudo-distance (ou "méthode de calage", comme définie dans [3]). On note enfin X_s la matrice de dimensions $n \times J$ des valeurs prises par les X_j sur s et $d_k = \dfrac{1}{\pi_k}$ le poids de sondage de l'unité $k \in s$. Le calage consiste à trouver les poids $w = (w_k)_{k \in s}$ vérifiant :

$$\begin{cases} \arg\min\limits_{w_k} \sum\limits_{k \in s} d_k G(\dfrac{w_k}{d_k}) \\ \text{s.c. } X_s' w = T_X \end{cases} \tag{1}$$

2.2 Exemple

Pour cet exemple, on considère une population de taille 300 et un échantillon par sondage aléatoire simple stratifié, où la variable d'intérêt est la fréquentation du cinéma.

2. Icarus Calibrates and Reweights Units in Samples
3. `https://cran.r-project.org/web/packages/icarus/index.html`

La population est décrite plus précisément dans le wiki du package [4]. Elle est également utilisée dans certains tests unitaires (voir partie 5).

Lors de l'installation de Icarus, les populations de test sont également chargées. Il est donc possible de visualiser l'échantillon de taille 15 après chargement du package :

```
R > library(icarus)
R > data_ex2
```

	id	service	categ	sexe	salaire	cinema	poids
1	a01	1	1	1	1000	1	10
2	a02	1	2	2	1100	2	10
3	a03	2	2	2	1500	4	10
4	a04	2	3	1	2300	15	10
5	a05	2	1	1	1000	2	10
6	a06	1	1	2	500	3	10
7	a07	2	2	2	1000	1	10
8	b01	1	3	2	2000	0	20
9	b02	1	1	1	2100	0	20
10	b03	2	2	1	2000	3	20
11	b04	2	1	2	3200	6	20
12	b05	1	1	2	1800	0	20
13	b06	1	2	1	2800	0	20
14	b07	1	3	1	1100	1	20
15	b08	2	1	2	2500	1	20

On connaît par ailleurs sur la population les nombres d'individus porteurs de chaque modalité pour les variables catégorielles "categ", "sexe" et "service" ainsi que le total de la variable quantitative "salaire". On peut créer la matrice des marges de calage ainsi :

```
## Calibration margins
mar1 <- c("categ",3,80,90,60)
mar2 <- c("sexe",2,140,90,0)
mar3 <- c("service",2,100,130,0)
mar4 <- c("salaire", 0, 470000,0,0)
marges <- rbind(mar1, mar2, mar3, mar4)
```

Comme dans Calmar2, il s'agit d'indiquer le nom et le type des variables de la table de données (en l'occurence "data_ex2") sur lesquelles le calage doit s'effectuer (première colonne de la matrice "marges"). La deuxième colonne renseigne le nombre de modalités de la variable catégorielle (on indique "0" lorsqu'il s'agit d'une variable quantitative).

4. https://github.com/haroine/icarus/wiki/Calibration

Les colonnes suivantes contiennent les totaux pour chaque modalité, classée par ordre alphanumérique. La matrice devant être carrée, chaque ligne de taille inférieure à 5 (la ligne correspondant à la variable "categ" contient le nom de la variable, le nombre de modalités (3), puis les 3 marges associées) est complétée par des zéros.

Il reste à effectuer le calage, par exemple avec la méthode logit et des bornes 0.4 et 2.1 :

```
R> wCalesLogit <- calibration(data=data_ex2, marginMatrix=marges,
                              colWeights="poids", method="logit",
                              bounds=c(0.4,2.1), description=TRUE)

#################### Summary of before/after weight ratios ###################
Calibration method : logit
 L bound : 0.4
 U bound : 2.1
    0%     1%    10%    25%    50%    75%    90%    99%   100%
0.4010 0.4018 0.4101 0.4909 0.8523 1.2445 1.7829 2.0770 2.0903
```

Le paramètre "description = TRUE" indique à Icarus qu'on souhaite afficher dans la sortie des statistiques résumant la procédure de calage. La première partie de ces statistiques (affichée plus haut pour notre exemple) concerne la distribution des "facteurs de calage" $g_k = \frac{w_k}{d_k}$. L'étude de la distribution des facteurs de calage permet d'apprécier l'étendue de la repondération qui est effectuée par le calage sur marges. La forme de cette distribution est parfois cruciale pour limiter le risque d'erreur d'estimation dans le cas d'enquêtes produisant des statistiques multiples (voir par exemple [5]). Icarus présente dans la sortie texte quelques quantiles de la distribution, et dans la sortie graphique en affiche le graphe de la densité estimée par noyau.

La seconde partie de la sortie (non retranscrite ici) indique l'écart entre l'estimation initiale et les marges de calage pour chaque variable du problème. Ceci permet de faciliter le déboguage des programmes de redressement, mais aussi parfois d'interpréter plus finement la repondération qui est effectuée.

3 Calage pénalisé

3.1 Principe

Le calage pénalisé consiste à relâcher la contrainte de marges et à l'intégrer dans le programme d'optimisation. Cette méthode a été étudiée entre autres dans [1, 6]. Elle permet de faciliter la convergence de la procédure et ainsi d'augmenter le nombre de variables pour lesquelles la valeur de l'estimation redressée est contrôlée tout en préservant

une distribution des facteurs de calage peu étendue. On conserve les notations de la partie 2. On note également $\widehat{T_{Xw}}$ le vecteur des estimateurs utilisant les poids w pour les variables auxiliaires X_j (vecteur des $\sum_{k\in s} w_k x_{jk}$). On se donne C est un vecteur de coût de taille le nombre de marges dans le programme de calage et $diag(C)$ désigne la matrice diagonale de dimensions $J \times J$ où les coefficients diagonaux sont les valeurs du vecteur C. Le programme de calage pénalisé s'écrit :

$$\min_{w_k} \sum_{k\in s} d_k G(\frac{w_k}{d_k}) + \lambda(\widehat{T_{Xw}} - T_X) diag(C)(\widehat{T_{Xw}} - T_X) \tag{2}$$

Il est à noter que l'on peut requérir un calage exact pour certaines marges en fixant un coût infini. Le paramètre λ est compris entre 0 et $+\infty$ et représente l'importance relative dans le programme de la distance aux poids initiaux et de l'écart aux marges des estimations redressées.

Tout comme pour le calage sur marges avec contraintes, on peut utiliser plusieurs distances G. On s'aperçoit néanmoins qu'introduire des distances bornées n'est pas utile pour limiter l'étendue des facteurs de calage (de repondération). En effet, si $\lambda \to +\infty$, alors le terme de coût est prépondérant : les contraintes de marges sont satisfaites en priorité, ce qui éloigne les facteurs de calage de 1. Si $\lambda \to 0$, alors le terme de distance est prépondérant, ce qui tend à rapprocher les facteurs de calage de 1.

Ainsi, Icarus choisit la plus grande valeur de λ telle que l'étendue de la distribution des facteurs de calage est inférieure à une valeur choisie par l'utilisateur (paramètre *gap*). Les distances bornées ne sont donc plus nécessaires. Ainsi, les deux distances disponibles pour le calage pénalisé sont la distance du khi-deux (" method='linear' ") et l'entropie relative (" method='raking' "). Dans le cas de la méthode linéaire, il existe une solution analytique. Pour la méthode du raking ratio, Icarus calcule la solution à l'aide de l'algorithme ICRS décrit par Bocci et Beaumont ([1]).

Le calage pénalisé a été récemment mis en œuvre en production à l'INSEE [5] en utilisant Icarus ([9, 4]). Il convient de noter qu'il n'existe pas à ce jour de moyen d'imposer une erreur relative d'estimation *a priori* pour les variables de cadrage. L'obtention d'une solution satisfaisante pour le statisticien se fait de manière empirique en jouant à la fois sur les paramètres de coût et de gap.

3.2 Exemple

On reprend l'exemple développé en partie 2.2. Cette fois ci, on souhaite resserrer les bornes du calage de manière à obtenir une étendue de distribution inférieure ou égale à 1.4. On utilise donc la méthode logit avec des bornes 0.6 et 2.0 :

5. Institut National de la Statistique et des Études Économiques, institut de statistique officielle en France

```
R > calibration(data=data_ex2, marginMatrix=marges,
            colWeights="poids", method="logit",
            bounds=c(0.6,2.0), popTotal = 230)
```

Malheureusement, ce programme de calage ne possède pas de solution, ce qu'indique Icarus :

```
Warning messages:
In calibAlgorithm(Xs, d, total, q, inverseDistance, updateParameters,  :
  No convergence
```

On choisit alors d'utiliser le calage pénalisé : on relâche la contrainte pour certaines marges afin d'obtenir une convergence avec l'étendue de distribution souhaitée. On décide que le calage doit être exact pour la variable "salaire", mais que les autres marges ont la même importance relative :

```
R > costs <- c(1,1,1,Inf)
R > calibration(data=data_ex2, marginMatrix=margins, colWeights="poids"
                , costs=costs, gap=1.4, description=TRUE, popTotal=230)
```

Après quelques itérations, Icarus affiche dans la log la sortie suivante :

```
Test with lambda = 0.00674285392857323
[1] 0.1949513
[1] 1.595005
Found lambda = 0.00674285392857323 ; count = 30

################### Summary of before/after weight ratios ###################
Calibration method : linear
Mean : 0.9661
    0%     1%    10%    25%    50%    75%    90%    99%   100%
0.1950 0.2434 0.5774 0.7712 0.9579 1.1268 1.4663 1.5819 1.5950

################### Comparison Margins Before/After calibration ###################
Careful, calibration may not be exact
$Total
Before calibration  After Calibration             Margin
               230                 230                230

$categ
  Before calibration After Calibration Margin
1                110             92.77     80
```

```
2                 70             84.08     90
3                 50             53.15     60

$sexe
  Before calibration After Calibration Margin
1                110             128.9    140
2                120             101.1     90

$service
  Before calibration After Calibration Margin
1                130            110.14    100
2                100            119.86    130

$salaire
Before calibration  After Calibration             Margin
            434000             470000             470000
```

Ainsi, le calage est bien effectué exactement pour la variable "salaire", et il subsiste une erreur des estimations pour les autres variables "de cadrage" (pour lesquelles le calage n'est pas effectué de façon exacte). On peut aussi remarquer que la distribution finale n'est pas centrée sur 1, ce qui n'est pas surprenant étant donné que l'échantillon est de taille très faible.

4 Calage sur bornes serrées

Les distances permettant d'inclure des bornes sont utiles lors d'un calage sur marges, car cela permet d'éviter de prendre le risque de repondérer trop fortement (notamment à la hausse) des unités influentes. De même, cela limite les risques d'explosion de la variance pour les estimations sur des domaines qui concentreraient un bon nombre de facteurs de repondération extrêmes ([10]).

Pour prolonger cette logique, on pourrait chercher les bornes L et U les plus rapprochées telles que l'équation de calage (1) possède toujours une solution. C'est le principe du calage sur bornes serrées. Le programme à résoudre est le suivant :

$$\begin{cases} \min\limits_{g\in\mathbb{R}^n} \left(\max\limits_{k\in[\![1,n]\!]} g_k - \min\limits_{k\in[\![1,n]\!]} g_k \right) \\ \qquad \text{s. c. } \widetilde{X}_s' g = T_X \ ; \ g \geq 0 \end{cases} \tag{3}$$

avec :

$$\widetilde{X}_s = diag(d)X_s$$
$$\forall k \in [[1, n]], g_k = \frac{w_k}{d_k} \text{ facteurs de calage}$$

Ce programme est linéaire et peut-être résolu par la méthode du simplexe (voir par exemple [13] ou [5]). Le meilleur algorithme à notre disposition (et implémenté dans Icarus) nécessite toutefois un nombre $\mathcal{O}(n^2)$ d'enregistrements en mémoire vive, ce qui peut être prohibitif pour des grandes tailles d'échantillon. Une méthode moins intensive en mémoire consiste à chercher ces bornes par dichotomie (voir [5]). Dans ce cas, la solution obtenue est sous-optimale, mais les résultats obtenus sont proches pour la plupart des échantillons.

Cette méthode par dichotomie est sélectionnée par défaut dans Icarus dès lors que $n \geq 10000$, afin de ne pas excéder la capacité en mémoire vive d'un PC de bureau typique en 2016. Il est toutefois possible (par exemple si l'on dispose d'un serveur de calcul) de forcer l'utilisation de l'algorithme du simplexe pour résoudre le programme (et donc aboutir à une solution optimale) en utilisant le paramètre "forceSimplex = TRUE".

Si les bornes serrées optimales ont été obtenues par la méthode du simplexe, un calage logit est testé avec les bornes obtenues. Il est toutefois rare que le calage converge en un temps acceptable avec ces bornes optimales. Pour cette raison, les bornes optimales sont tronquées avant que le calage soit testé. Si la convergence n'est toujours pas obtenue, une solution convergente est cherchée par dichotomie entre les bornes obtenues par méthode linéaire et les bornes obtenues par la méthode du simplexe. La précision avec laquelle la solution finale est demandée peut être réglée avec le paramètre *precisionBounds*.

Le calage sur bornes serrées se distingue par la forme de la distribution des facteurs de calage, qui présente fréquemment une forme "en U". Une étude statistique de l'estimation par calage sur bornes serrées (en particulier quand la distribution des facteurs de calage présente cette forme typique) est effectuée dans [5].

5 Tests unitaires et développements futurs

Icarus ayant vocation a être largement utilisé en production dans les instituts de statistique publique, il convient de limiter au maximum le risque d'une erreur provenant de l'implémentation. Ainsi, de nombreux tests unitaires ont été intégrés au processus de compilation de Icarus, couvrant une majorité des paramètres utilisables de la fonction "calibration". Ces ajouts ont été facilités par l'excellent package test_that de Hadley Wickham ([14]).

L'ambition pour le futur d'Icarus est de maintenir le développement actif, en étant particulièrement attentifs aux remontées d'utilisateurs qui pourraient être faites [6]. Il est également prévu d'ajouter des fonctions pour le calage simultané ([2]) et le calage sur variables non linéaires ([7]), ainsi que des fonctions facilitant la correction de la non-réponse par repondération (notamment par GRH [7]).

Références

[1] C. Bocci et J.-F. Beaumont. "Another look at ridge calibration". In : *Metron* 66.1 (2008), p. 5–20.

[2] M. Cordier-Villoing et O. Sautory. "Redressement de la non-réponse et calage dans les enquêtes couplées". In : *Actes des Journées de Méthodologie Statistique, Paris* (2012).

[3] J.-C. Deville et C.-E. Särndal. "Calibration estimators in survey sampling". In : *Journal of the American statistical Association* 87.418 (1992), p. 376–382.

[4] E. Gros. "Note de bilan relative aux calculs de pondérations dans l'enquête nationale sur les ressources des jeunes". In : *Document de travail INSEE* (2015).

[5] E. Gros et A. Rebecq. "Étude du calage sur bornes minmales". In : *Actes du 9ème colloque francophone sondages* (2016).

[6] F. Guggemos et Y. Tillé. "Penalized calibration in survey sampling : Design-based estimation assisted by mixed models". In : *Journal of statistical planning and inference* 140.11 (2010), p. 3199–3212.

[7] É. Lesage. "Calage non linéaire". In : *Actes Xèmes Journées Methodologie Statistique* (2009).

[8] T. Lumley et al. "Analysis of complex survey samples". In : *Journal of Statistical Software* 9.1 (2004), p. 1–19.

[9] A. Rebecq et C. Favre-Martinoz. "Équilibrage des poids des ZAE pour le plan de sondage OCTOPUSSE". In : *Document de travail INSEE* (2016).

[10] G. Roy et A. Vanheuverzwyn. "Redressement par la macro CALMAR : applications et pistes d'amélioration". In : *Traitements des fichiers d'enquête* (2001), p. 31–46.

[11] O. Sautory. "Calmar 2 : A new version of the Calmar calibration adjustment program". In : *Proceedings of Statistics Canada Symposium*. 2003.

6. La page privilégiée pour rapporter un bug ou une demande de fonctionnalité est la page "issues" de GitHub : `https://github.com/haroine/icarus/issues`

7. Groupes de Réponse Homogène

[12] O. Sautory. “La macro Calmar. Redressement d’un échantillon par calage sur marges”. In : *Document F9310, DSDS, INSEE* (1993).

[13] C. Vanderhoeft. “Generalized calibration at Statistics Belgium”. In : *Technical Report 3, Statistics Belgium WorkingPaper* (2001).

[14] H. Wickham. “testthat : Get started with testing”. In : *The R Journal* 3.1 (2011), p. 5–10.