Communauté Economique et Monétaire de l'Afrique Centrale
(CEMAC)
Institut Sous regional de Statistique et d'Economie Appliquée
(ISSEA)
Organisation Internationale

B.P 294, Yaoundé (République du Cameroun)
Tél. (237) 22-22-01-34 **Fax**. (237) 22-22-95-21
Email: isseacemac@yahoo.fr **web**: www.issea-cemac.org

PROJET DE LOGICIELS STATISTIQUES II (IAS3-2021)

Introduction

Le rééchantillonnage est une méthode empirique utilisée lorsque la distribution théorique d'une variable aléatoire $\boldsymbol{X}$ ou d'un paramètre $\boldsymbol{\theta}$ n'est pas connue et que nous souhaitons malgré tout fournir des indicateurs statistiques sur eux.

Le principe du rééchantillonnage est de tirer des sous-échantillons, selon certaines propriétés, dans l'échantillon observé et d'étudier la nouvelle variable aléatoire conçue soit directement depuis les sous-échantillons soit au travers d'un modèle ou des indicateurs statistiques que nous puissions en tirer. Cet outil permet alors de considérer des intervalles de variations pour $\boldsymbol{X}$ ou $\boldsymbol{\theta}$ et ainsi estimer la moyenne et l'écart-type dans le but de construire l'intervalle de confiance associé.

Le ***Jack-knife*** est la première méthode de rééchantillonnage mise sur pied. Elle a été conçue par le statisticien britannique **Maurice Quenouille** en 1950 et étendue par **John Wilder Tuckey** en 1958. De nos jours, la plus célèbre des méthodes de rééchantillonnage de par la simplicité de son algorithme, sa robustesse et son adaptation à plusieurs domaines de la statistique est le ***bootstrap***. Le bootstrap est l'œuvre des travaux du Statisticien américain **Bradley Efron**. le *Jack-knife* reste une méthode utile dans certains cas particuliers notamment pour le problème des estimateurs qui sont suspectés d'un biais.

Objectif du projet

L'objectif du projet est d'écrire une fonctions R qui permet de simuler par rééchantillonnage un intervalle de confiance d'un paramètre calculé sur un ensemble de données. La fonction sera testée sur la simulation d'un intervalle de confiance de la mesure de pauvreté multidimensionnelle suivant la Méthode AF (Sabina Alkire & James Foster,2009) à partir d'une matrice des réalisations d'une population. En effet, la plupart des mesures de la pauvreté ne suivent pas une distribution théorique connue, et souvent le seul moyen d'en dégager des indicateurs statistiques de robustesse est de de faire appels aux méthodes de rééchantillonnage ou de simulation.

Le projet est composé de deux (02) fonctions :

- *La fonction qui calcule la mesure de pauvreté multidimensionnelle suivant la méthode AF ;*
- *La fonction de simulation de l'intervalle de confiance d'un paramètre calculé à partir d'une fonction et des données d'une population.*

lucdimai@yahoo.fr / lucdimai@gmail.com

Partie 1 : La fonction de simulation de l'intervalle de confiance

Nom de la fonction : **simulerIC**

Objets en entrée de la fonction :

1. **database**, un *dataframe* ou un *vecteur* qui contient les données à utiliser;
2. **typeSondage**, un objet de type *character* qui ne peut prendre que deux valeurs : **SAS** pour sondage aléatoire simple et **SAT** pour sondage aléatoire stratifié. **typeSondage** permet de préciser la méthode de tirage par laquelle l'échantillon des données de **database** été obtenu à partir de la population ;
3. **strate,** un entier qui indique le numéro de la variable de stratification dans l'ensemble **database.** Si **strate** est égal à zéro alors il n'y a pas de variable de stratification dans **database.** ;
4. **FUN,** le nom de la fonction qui permet de calculer le paramètre à estimer à partir des données de database. La fonction **FUN** est une fonction qui doit être présente dans l'environnement de la session R pour qu'elle puisse être utilisée par **simulerIC**. Cette fonction prend en entrée toutes les données de **database** à l'exception de la variable de stratification au cas où celle-ci est présente dans l'ensemble de données.
5. ***methoSim*,** un objet de type *character* qui ne peut prendre que deux valeurs : **jackknife** et **bootstrap**. ***methodSim*** permet de choisir la méthode de rééchantillonnage à utiliser ;
6. **times**, le nombre de simulations à effectuer. La valeur par défaut de times est 1000. *times* n'est utilisé qu'au cas où la méthode de simulation choisie est le *bootstrap* ;
7. **er,** un réel entre **0** et **1** qui précise qu'on simule un intervalle de confiance à ***(1-er) x100* %.** Par défaut **er=0.05** ce qui correspond à un niveau de confiance de **95% ;**
8. Tout paramètre nécessaire à l'utilisation de la fonction **FUN**.

Objet en sortie de la fonction : **resultat**, un objet de classe ***List*** qui contient 07 éléments :

1. les données utilisées ;
2. le nom de la méthode de simulation utilisée ;
3. le nom de la fonction utilisée pour calculer le paramètre cible ;
4. la valeur du paramètre calculée sur les données ;
5. un vecteur de 02 éléments pour les bornes de l'intervalle de confiance ;
6. le niveau de confiance de l'IC ;
7. le vecteur de toutes les valeurs du paramètre calculées à chaque rééchantillonnage.

Méthodologie de simulation de l'intervalle de confiance par la méthode bootstrap :

1. Tirer avec remise un échantillon de la même taille que le nombre d'individus dans ***database***. S'il y a une variable de stratification, subdiviser d'abord ***database*** en sous bases selon les modalités de la variable strate. Ensuite, faire un tirage avec remise d'un sous-échantillon de taille identique à la sous base dans chaque sous-base et rassembler tous les sous-échantillons pour former le nouvel échantillon.
2. Appliquer la fonction **FUN** à cet échantillon ;

lucdimai@yahoo.fr / lucdimai@gmail.com

3. Répéter les étapes de 1 à 2 ***times*** fois en stockant toutes les valeurs calculées dans un vecteur de taille ***times ;***
4. Déterminer les quantiles d'ordre $\mathbf{er/2}$ et $\mathbf{1\text{-}er/2}$ de la distribution des valeurs calculées. L'on obtient ainsi notre intervalle de confiance simulé à niveau de confiance $\mathbf{1\text{-}er}$.

Méthodologie de simulation de l'intervalle de confiance par la méthode jack-knife d'ordre 1 :

1. Pour chaque étape i de la simulation allant de 1 à la taille du nombre d'individus de ***database*** :
   8. Prendre l'échantillon E_{-i} en éliminant l'individu i ;
   9. Appliquer la fonction **FUN** à E_{-i} .
2. Stocker toutes les valeurs calculées dans un vecteur de taille *nombre d'individus de* ***database.***
3. Déterminer les quantiles d'ordre $\mathbf{er/2}$ et $\mathbf{1\text{-}er/2}$ de la distribution des valeurs calculées. L'on obtient ainsi notre intervalle de confiance simulé à niveau de confiance $\mathbf{1\text{-}er}$.

Partíe 2 : La fonctíon de calcul de la mesure de pauvreté suívant la méthode AF

Sabina Alkire et James *Foster* ont proposé en 2009 dans un article intitulé « *Counting and Multidimensional poverty Measurement* » une méthodologie d'estimation de la pauvreté multidimensionnelle qui repose sur une étape d'indentification des individus pauvres à double seuil et une étape d'agrégation des informations sur les pauvres qui s'appuie sur les mesures FGT (Foster-Greer-Thorbecke) (Foster, et al., 1987) pour déduire des classes de mesures M_α. Pour ce projet nous nous intéressons à la mesure M_0.

Supposons une matrice X de n lignes et d colonnes décrivant les réalisations d'une population de taille n suivant d variables cardinales ou catégorielles. x_{ij} est la valeur prise par la variable j pour l'individu i. Supposons également $(z_1, \dots, z_j, \dots, z_d)$ le vecteur des seuils de réalisation par indicateur en deçà desquelles un individu est considéré en situation de privation dans un indicateur et $(\omega_1, \dots, \omega_j, \dots, \omega_d)$ le vecteur qui représente les poids des différents indicateurs tel que $\sum\omega_j = d$. La première étape consiste à déterminer la matrice de privation pondérée g^0 définie par :

$$g_{ij}^0 = \begin{cases} 0 & si\ x_{ij} \geq z_j \\ \omega_j & sinon \end{cases}$$

Par la suite les privations pondérées de chaque individu sont additionnées pour obtenir le vecteur d'intensité de privation $c = (c_1, \dots, c_i, \dots, c_n)$. Notons $k \in [1\,;d]$ le seuil d'intensité de privation qui correspond au plafond d'intensité de privation qu'un individu ne doit pas atteindre au risque de se trouver en situation de pauvreté. L'identification des personnes pauvres se fait à travers la fonction d'identification définie par :

$$\rho_k : \mathbb{R}_+^d \times \mathbb{R}_{++}^d \to \{0\,;1\}$$
$$(x_i; z) \mapsto \begin{cases} 1\ si\ c_i \geq k \\ 0 \quad sinon \end{cases}$$

Soit q le nombre de pauvres de la population. L'incidence de la pauvreté dans la population est donnée par :

$$H = \frac{q}{n}$$

En raison de l'axiome (propriété désirable) de concentration, les mesures de pauvreté s'appuient uniquement sur les informations des personnes pauvres. Notons par $c(k)$ le vecteur d'intensité de privations censuré définie par $c_i(k) = c_i \times \rho_k(x_i; z)$ et l'on considérera $g^0(k)$ comme la matrice des privations censurée ou encore la matrice de privation qui ne contient que les intensités de privation des personnes pauvres et où les intensités de

lucdimai@yahoo.fr / lucdimai@gmail.com

privation des non pauvres ont été ramenés à zéro. Ainsi, $\frac{c_i(k)}{d}$ est la proportion d'intensité de privation expérimentée par l'individu i . La proportion moyenne d'intensité de privation parmi les pauvres se définit ainsi par :

$$A = \frac{\sum_i c_i(k)}{qd}$$

A est une mesure de la sévérité de la pauvreté. Dès lors, la mesure de pauvreté ajusté suivant la méthode AF se définit par :

$$M_0 = H \times A = \frac{\sum_i c_i(k)}{nd} = \frac{\sum_i \sum_j g_{ij}^0(k)}{nd}$$

Consigne :

Vous allez écrire une fonction nommée **povAF** qui a comme paramètre d'entrée la matrice des réalisations X (X peut être un *dataframe* ou une *matrice*) de la population qui contient uniquement les indicateurs sur les réalisations des individus permettant d'appréhender leur bien-être et qui fournit en sortie un vecteur de trois éléments qui contient la mesure M_0 calculée et ses composantes A et H. La fonction prend aussi entrée les vecteurs **z** pour les seuils et **w** pour les pondérations. Ces 02 vecteurs ont pour taille le nombre de colonnes de $\boldsymbol{X}$. Le dernier paramètre de la fonction **povAF** est le réel $k \in [0\,;d]$ qui représente le seuil d'intensité de privation tel que décrit plus haut. d est le nombre de colonnes de la matrice X.

NB : *pour toutes les fonctions qui seront écrites, des tests préalables doivent être effectués en entrée pour s'assurer de la conformité du type des paramètres et éventuellement produire des messages d'erreurs conséquents.*

Partie 4 : Test des différentes fonctions

Un ensemble de données pour tester vos fonctions est fournie dans le fichier ***dataTest.csv*** C'est un ensemble de données de d indicateurs sur les réalisations d'une population. Cet ensemble contient une variable de stratification.

1. Calculer M_0 et ses composantes A et H à partir de cet ensemble de données. Vous utiliserez $z = (1, \dots, 1)$ et $w = (1, \dots, 1)$ et enfin $k = \frac{2}{5}d$, d est le nombre d'indicateurs.
2. Simuler un intervalle de confiance 95% pour la valeur de M_0 calculée avec 2000 rééchantillonnages par la méthode bootstrap ;
3. Simuler un intervalle de confiance 95% pour la valeur de M_0 calculée par la méthode *Jack-knife* ;

Le package final du projet à fournir par chaque groupe est un dossier contenant 02 fichiers :

4. **un fichier d'extension *.r* qui contient les scripts de vos fonctions avec les noms des membres du groupe ;**
5. **un fichier Excel qui contient les résultats finaux obtenus lors du test.**

Une prime de notation sera accordée aux projets qui sauront minimiser le temps d'exécution de leurs scripts avec une bonne utilisation des possibilités du langage R.

Vous avez jusqu'au vendredi 12 mars 2021 délai de rigueur pour envoyer votre projet par mail !

lucdimai@yahoo.fr / lucdimai@gmail.com