

Estimation de la variance à l'aide des poids de bootstrap

Guide de l'utilisateur du programme BOOTVARF.SAS pour utilisation avec le fichier fictif de l'ENSP 1994-95

1. Méthode du bootstrap

Le plan d'échantillonnage pour l'Enquête nationale sur la santé de la population (ENSP) est complexe. On utilise donc une méthode de rééchantillonnage pour calculer des estimations de la variance. Pour l'ENSP, on a utilisé la méthode du bootstrap.

Cette méthode consiste à diviser les enregistrements sur le fichier de données en sous-groupes (ou répétitions) et à déterminer la variance entre les estimations pour chaque répétition. Les répétitions sont formées de la manière suivante: à l'intérieur de chaque strate, on choisit un échantillon aléatoire simple (ÉAS), avec remise, de $n-1$ grappes parmi les n grappes de la strate. Pour chaque échantillon ÉAS, le facteur de pondération est recalculé pour chaque enregistrement dans la strate. Ce poids recalculé correspond au premier poids bootstrap. Ce processus est répété B fois afin de former un ensemble de B poids bootstrap pour chaque enregistrement sur le fichier de données.

2. Poids bootstrap pour l'ENSP

Pour l'ENSP, deux ensembles de poids bootstrap ont été créés. L'ensemble BD5H35 est utilisé avec le fichier fictif DUMYH35 et l'ensemble BD5H356 avec le fichier fictif DUMYH356. Ils contiennent tous deux 500 poids bootstrap pour chaque enregistrement.

Les poids bootstrap sont utilisés pour obtenir des estimations de variance précises pour des statistiques telles que des totaux et ratios, et pour des analyses plus complexes comme des régressions. Les mêmes règles de confidentialité et seuils de diffusion s'appliquent pour les estimations de variance provenant du bootstrap.

L'ensemble de poids approprié doit être utilisé pour une estimation valide de la variance d'une estimation donnée. Par exemple, si on veut calculer l'estimation d'un total en utilisant le fichier DUMYH35, on suit les étapes suivantes:

- A) On calcule d'abord une estimation du total en utilisant le poids final WT54.
- B) On calcule ensuite une estimation du total voulu en utilisant chacun des 500 poids bootstrap contenus dans le fichier BD5H35 comme pondération de l'échantillon. On obtient alors 500 estimations du total en question.
- C) Finalement, on calcule la variance de ces 500 estimations. Cette variance est l'estimation de

la variance du total calculé en A).

3. Programme BOOTVARF.SAS

Le programme SAS BOOTVARF.SAS est fourni avec les ensembles de données comme exemple pour calculer l'estimation de la variance. Il peut être utilisé par les utilisateurs ou ceux-ci peuvent créer leur propre programme. Une copie du programme apparaît à la fin de cette section.

Le programme BOOTVARF.SAS calcule une estimation (par exemple, un total) et une estimation de la variance de cette dernière. Il est toutefois conseillé de calculer l'estimation désirée (dans ce cas-ci, le total) à l'aide d'un autre programme pour vérifier que le programme BOOTVARF.SAS calcule bien l'estimation voulue.

Dans cette section, on décrit comment préparer les données pour le programme BOOTVARF.SAS et comment l'utiliser. Deux exemples sont donnés ainsi que le temps requis pour soumettre le programme (ces programmes ont été exécutés en utilisant un Pentium Pro 500 MHz, avec 128 MB de RAM).

3.1 Définition des variables pour l'analyse

Le programme BOOTVARF.SAS permet de calculer des estimations de variance pour des totaux, des rapports, des différences entre des rapports, et des paramètres de régression linéaire et logistique. Dans un premier temps, les données doivent être préparées et sauvegardées dans un fichier SAS pour réaliser les analyses voulues. Pour des estimations de totaux, rapports et différences entre des rapports, les estimations sont calculées en sommant le poids des enregistrement qui ont la caractéristique d'intérêt. Une variable indicatrice doit être créée pour identifier ces enregistrements (voir exemple 1A). Pour les analyses de régression, les variables à analyser doivent être définies de la façon requise par SAS (voir exemple 1B).

Une variable supplémentaire (REGION) peut être nécessaire si les analyses sont réalisées pour des sous-groupes géographiques d'enregistrements sur le nouveau fichier de données créé pour le programme BOOTVARF.SAS. Par exemple, si le nouveau fichier d'analyse contient tous les enregistrements des personnes âgées de 12 ans et plus au Canada, l'ensemble des enregistrements représente un groupe qui correspond au Canada. Si les analyses sont faites au niveau du Canada, la variable REGION n'est pas nécessaire. Toutefois, si les analyses doivent être réalisées au niveau des provinces, alors la variable REGION est nécessaire. Cette variable doit être créée par l'utilisateur et prendra, pour chaque enregistrement, la valeur de la région géographique de cet enregistrement, dans ce cas-ci, la province. De même, si le nouveau fichier d'analyse contient tous les enregistrements des personnes âgées de 12 ans et plus dans les provinces de l'Ontario, du Manitoba et de l'Alberta, la variable REGION sera nécessaire si les analyses sont faites au niveau des provinces ou encore au niveau des régions sanitaires. Ici aussi, la variable prendra la valeur de la province.

L'exemple suivant démontre comment définir les variables qui seront ensuite utilisées pour calculer des estimations de variance à l'aide du programme BOOTVARF.SAS.

Exemple 1: Définition des variables pour l'analyse

Exemple 1A: Totaux et rapports

```

/*****
/*
/*      UTILISE LE FICHIER TRANSVERSAL DE LA COMPOSANTE GÉNÉRALE
/*      DE 1994 (DUMYH35.TXT)
/*      POUR CALCULER LE NOMBRE DE PERSONNES DIABÉTIQUES
/*      POUR LES GROUPES D'ÂGES 12-17, 18-24, 25-44, ET 45 ET PLUS
/*      ET POUR LES HOMMES, FEMMES, ET LES DEUX,
/*      PAR PROVINCE
/*
*****/

libname in1 'c:\data';

proc datasets kill;
run;

proc format;
  value fsex      1='hommes'
                  2='femmes';

  value fage      1='12-17'
                  2='18-24'
                  3='25-44'
                  4='45+';

  value fdiab     1='Oui'
                  0='Non';
run;

data in1.cross;
%let datafid='c:\data\dumyh35.txt';
%include 'c:\data\h35_in.sas';
keep region diab total dhc4_sex hommes femmes
    agegrp age1217 age1824 age2544 age45 magegrp fagegrp
    mage1217 mage1824 mage2544 mage45 fage1217 fage1824 fage2544 fage45
    mdiab fdiab diab1217 diab1824 diab2544 diab45
    mdia1217 mdia1824 mdia2544 mdia45 fdi a1217 fdi a1824 fdi a2544 fdi a45
    wt54 total realukey personid;

region=prc4_cur;

if ccc4_1j=1 then diab=1;
else if ccc4_1j=2 then diab=0;
else diab=.;

if diab=. then delete;

/*****
/* définit les domaines d'intérêt avec des
/* variable indicatrices (0/1)
*****/

total=1;

/* sexe */

if dhc4_sex=1 then hommes=1;
else hommes=0;

if dhc4_sex=2 then femmes=1;
else femmes=0;

/* groupe d'âge */

if dhc4_age < 18 then agegrp=1;
else if dhc4_age < 25 then agegrp=2;
else if dhc4_age < 45 then agegrp=3;
else agegrp=4;

if agegrp=1 then age1217=1; else age1217=0;
if agegrp=2 then age1824=1; else age1824=0;
if agegrp=3 then age2544=1; else age2544=0;
if agegrp=4 then age45=1; else age45=0;

```

```

/* groupe âge-sexe */
if agegrp=1 then do;
  if dhc4_sex=1 then magegrp=1;
  else if dhc4_sex=2 then fagegrp=1;
end;
else if agegrp=2 then do;
  if dhc4_sex=1 then magegrp=2;
  else if dhc4_sex=2 then fagegrp=2;
end;
else if agegrp=3 then do;
  if dhc4_sex=1 then magegrp=3;
  else if dhc4_sex=2 then fagegrp=3;
end;
else if agegrp=4 then do;
  if dhc4_sex=1 then magegrp=4;
  else if dhc4_sex=2 then fagegrp=4;
end;

if magegrp=1 then mage1217=1; else mage1217=0;
if magegrp=2 then mage1824=1; else mage1824=0;
if magegrp=3 then mage2544=1; else mage2544=0;
if magegrp=4 then mage45=1; else mage45=0;

if fagegrp=1 then fage1217=1; else fage1217=0;
if fagegrp=2 then fage1824=1; else fage1824=0;
if fagegrp=3 then fage2544=1; else fage2544=0;
if fagegrp=4 then fage45=1; else fage45=0;

/* diabète*sexe */
if diab=1 then do;
  if dhc4_sex=1 then mdiab=1; else mdiab=0; /* hommes diabétiques */
  if dhc4_sex=2 then fdiab=1; else fdiab=0; /* femmes diabétiques */
end;
else if diab=2 then do;
  mdiab=0;
  fdiab=0;
end;

/* diabète*groupe d'âge */
if diab=1 then do;
  if age1217=1 then diab1217=1; else diab1217=0;
  if age1824=1 then diab1824=1; else diab1824=0;
  if age2544=1 then diab2544=1; else diab2544=0;
  if age45=1 then diab45=1; else diab45=0;
end;
else do;
  diab1217=0;
  diab1824=0;
  diab2544=0;
  diab45=0;
end;

/* diabète*sexe*groupe d'âge */
if mdiab=1 then do;
  if age1217=1 then mdia1217=1; else mdia1217=0;
  if age1824=1 then mdia1824=1; else mdia1824=0;
  if age2544=1 then mdia2544=1; else mdia2544=0;
  if age45=1 then mdia45=1; else mdia45=0;
end;
else do;
  mdia1217=0;
  mdia1824=0;
  mdia2544=0;
  mdia45=0;
end;

if fdiab=1 then do;
  if age1217=1 then fdia1217=1; else fdia1217=0;
  if age1824=1 then fdia1824=1; else fdia1824=0;
  if age2544=1 then fdia2544=1; else fdia2544=0;
  if age45=1 then fdia45=1; else fdia45=0;
end;
else do;
  fdia1217=0;
  fdia1824=0;
  fdia2544=0;
  fdia45=0;
end;

proc freq data=in1.cross;
format agegrp fage. magegrp fage. fagegrp fage. diab fdiab.;
table region*(diab diab2544 mdia45 fdia45)
       region*diab*total region*mdia45*mage45 region*diab45*age45 ;
weight wt54;

```

```

title 'Nombre de diabétiques par groupe d'âge et région';
run;

```

(L'exécution de ce programme a pris environ 2 minutes.)

Exemple 1B: Régression logistique

```

/******
/*
/*          UTILISE LE FICHIER TRANSVERSAL DE LA COMPOSANTE SANTÉ          */
/*          DE 1994 (DUMH356.TXT)                                          */
/*          POUR ETUDIER LA RELATION ENTRE LE LES PROBLÈMES DE SNATÉ      */
/*          CHRONIQUES, L' AGE ET LE SEXE                                  */
/*          */
/******

libname in1 'c:\data';

data in1.dumyh356;
%let datafid="c:\data\dumyh356.txt";
%include "c:\data\h356_in.sas";
keep realukekey personid dhc4_age dhc4_sex ccc4_1v
age1217 age1820 age2124 age2534 age3544 age4554 age5564
age6574 age7584 age85 femmes wt64;
if dhc4_age >= 12;

/* Variables indicatrices pour les groupes d'âge */

age1217=0;
age1820=0;
age2124=0;
age2534=0;
age3544=0;
age4554=0;
age5564=0;
age6574=0;
age7584=0;
age85=0;

if 12 <= dhc4_age <=17 then age1217=1;
if 18 <= dhc4_age <=20 then age1820=1;
if 21 <= dhc4_age <=24 then age2124=1;
if 25 <= dhc4_age <=34 then age2534=1;
if 35 <= dhc4_age <=44 then age3544=1;
if 45 <= dhc4_age <=54 then age4554=1;
if 55 <= dhc4_age <=64 then age5564=1;
if 65 <= dhc4_age <=74 then age6574=1;
if 75 <= dhc4_age <=84 then age7584=1;
if dhc4_age>=85 then age85=1;

femmes=0;
if dhc4_sex=2 then femmes=1;

/* Valeurs manquantes */

if ccc4_1v >2 then ccc4_1v=.;
if ccc4_1v=2 then ccc4_1v=0;

run;

proc logistic data=in1.dumyh356 descending;
model ccc4_1v = age1820 age2124 age2534 age3544
age4554 age5564 age6574 age7584 age85 femmes;
weight wt64;
title 'Profil des gens avec problèmes de santé chronique';
run;

```

(L'exécution de ce programme a pris environ 1 minute.)

3.2 Utilisation du programme BOOTVARF.SAS

Une fois le nouveau fichier d'analyse créé, le programme BOOTVARF.SAS peut être exécuté. L'utilisateur doit toutefois compléter le programme en spécifiant des paramètres et en indiquant la

liste des variables à analyser. Le programme BOOTVARF.SAS est divisé en quatre sections.

À la **section 1**, l'utilisateur doit spécifier:

- le nombre de poids bootstrap utilisés (500 pour chacun des deux ensembles: DUMYH35 et DUMYH356),
- les répertoires contenant le nouveau fichier d'analyse (créé par l'utilisateur, comme dans l'exemple 1), l'ensemble de poids bootstrap, et le fichier qui contiendra les résultats,
- le nom du nouveau fichier d'analyse,
- le nom du fichier contenant l'ensemble de poids bootstrap (le fichier doit être de format SAS),
- les trois paramètres qui indiquent si les analyses sont faites à un niveau géographique.

À la **section 2**, des changements doivent être faits seulement si l'usager désire analyser un modèle linéaire généralisé. Ces changements sont pour la macro définie pour le modèle linéaire généralisé. Cette macro doit être modifiée par l'utilisateur selon l'analyse voulue.

À la **section 3**, l'utilisateur doit spécifier la liste des variables à conserver pour les analyses. Le programme BOOTVARF.SAS est exécuté assez rapidement lorsqu'on calcule des estimations de variance pour des totaux, des rapports, et des différences entre des rapports. Cependant, plus de temps est requis dans le cas des estimations de variance pour les paramètres d'une régression. Afin de limiter le temps d'exécution, il est donc recommandé de ne conserver que les variables nécessaires aux fins de l'analyse voulue.

En plus des variables d'analyse, deux variables doivent être conservées sur le fichier: les variables REALUKEY et PERSONID. Ces variables sont nécessaires pour faire l'appariement entre le fichier de données et le fichier contenant l'ensemble de poids bootstrap durant l'exécution du programme. Il est à noter que le programme BOOTVARF.SAS lit deux fichiers: les données et les poids bootstrap, et ces deux fichiers doivent être des fichiers SAS.

Finalement, à la **section 4**, l'utilisateur doit indiqué quelles analyses seront exécutées et pour quelles variables.

Les exemples suivant indiquent en caractères ***italiques*** ce que l'utilisateur doit spécifier dans le cas d'estimations de variance pour des totaux et des rapports (exemple 2A), et pour des paramètres de régression logistique (exemple 2B). Il est à noter que seuls des extraits du programme où des changements sont nécessaires sont présentés. Il faut cependant soumettre le programme au complet.

Exemple 2: Programme BOOTVARF.SAS

Exemple 2A: Totaux et rapports

```
.
.
.

/*****/
*** Section 1: Déclaration des variables macros *****/
/*****/

%let R=01;                      /* paramètre du bootstrap moyen (fixe) */

/*****/
*** L'UTILISATEUR DOIT INDIQUE LE NOMBRE DE POIDS BOOTSTRAP À UTILISER, *****/
*** DIVISÉ PAR 100 *****/
*** %let B=5; pour les fichiers de poids bootstrap BD5H35 et BD5H356 *****/
/*****/

%let B=5;                      /* nombre de poids de bootstrap Bx100 */

/*****/
*** L'UTILISATEUR DOIT INDiquer LE RÉPERTOIRE CONTENANT LE FICHIER *****/
*** D'ANALYSE (in1) ET LE FICHIER CONTENANT LES POIDS DE BOOTSTRAP *****/
*** (in2). LES FICHIERs DE SORTIES SERONT SAUVEGARDEs DANS LE *****/
*** RÉPERTOIRE SPÉCIFIÉ POUR in3. *****/
/*****/

libname in1 "c:\data";
libname in2 "c:\bootstrp";
libname in3 "c:\output";

/*****/
*** L'UTILISATEUR DOIT INDiquer LE NOM DU FICHIER SAS CONTENANT LES VARIABLES *****/
*** D'ANALYSE. *****/
*** %let Mfile=xxx; *****/
/*****/

%let Mfile=in1.cross;          /* Nom du fichier SAS principal */

/*****/
*** L'UTILISATEUR DOIT INDiquer LE NOM DU FICHIER SAS CONTENANT LES POIDS DE *****/
*** BOOTSTRAP. SI LE FICHIER SAS N'EXISTE PAS, IL DOIT ÊTRE CRÉÉ. *****/
*** %let bsamp=bd5h35; (à utiliser avec le fichier DUMH35.TXT) OU *****/
*** %let bsamp=bd5h356; (à utiliser avec le fichier DUMH356.TXT) *****/
/*****/

data in2.bd5h35;
    %let datafid="c:\bootstrp\bd5h35.txt";
    %include "c:\bootstrp\boot_in.sas";
run;

%let bsamp=in2.bd5h35;          /* fichier SAS des poids bootstrap */

/*****/
*** L'UTILISATEUR DOIT SPÉCIFIER LA VALEUR DES PARAMÈTRES by, *****/
*** wo ET kp. CEUX-CI INDIQUENT SI L'ANALYSE EST FAITE AU *****/
*** NIVEAU RÉGIONAL OU POUR L'ENSEMBLE DU FICHIER. *****/
/*****/
*** SI L'ANALYSE EST FAITE POUR L'ENSEMBLE DU FICHIER, ALORS *****/
*** %let by=*; *****/
*** %let wo=*; *****/
*** %let kp=*; *****/
/*****/
*** SINON, SI L'ANALYSE EST FAITE AU NIVEAU RÉGIONAL, ALORS *****/
*** %let by=*; *****/
*** %let wo=*; *****/
*** %let kp=region; *****/
/*****/

%let by=;
%let wo=*;
%let kp=region;

data time;
    format start datetime16.;
    start=datetime();
    output;
run;
```

```

/*****
*** Section 2: Déclaration des macros ****
*****/

.
.
.

*****,
* Section 3: Programme principal *,
*****,

/* Lit le fichier principal */

/*****
*** À CAUSE DU GRAND NOMBRE D'OBSERVATIONS, SEULES LES VARIABLES ****
*** REQUISES POUR FAIRE LES ANALYSES ET LES VARIABLES, ****
*** "REALUKEY" ET "PERSONID" DEVRAIENT ÊTRE CONSERVÉES DÉSORMAIS. ****
*****/

data nphs (index=(id=(realukey personid)));
  set &MFile (keep= realukey personid region diab total dhc4_sex
               hommes femmes agegrp age1217 age1824 age2544 age45
               magegrp fagegrp mage1217 mage1824 mage2544 mage45
               fage1217 fage1824 fage2544 fage45 mdiab fdiab
               diab1217 diab1824 diab2544 diab45 mdia1217 mdia1824
               mdia2544 mdia45 fdia1217 fdia1824 fdia2544 fdia45
               wt54 total);

run;

/* Lit les poids de bootstrap */
/* FWGT est le même poids que WT54 ou WT64, selon le fichier analysé */

data bsamp (index=(id=(realukey personid)));
  set &bsamp;
  keep fwgt realukey personid bsw1- bsw&B. 00;
run;

/* Apparie le fichier principal et les poids bootstrap */

&by data bs_nphs (index=(region));
&wo data bs_nphs;
      merge nphs (in=in1) bsamp (in=in2);
      by realukey personid;
      if in1;
run;

proc datasets library=work;
  delete nphs bsamp;
run;

/*****
*** Section 4: Énoncés pour soumettre les macros ****
*****/

/*****
*** Les estimations de variance sont calculées à l'aide des macros SAS définies ***
*** plus tôt dans le programme. Ces macros SAS peuvent être soumises durant ***
*** l'exécution du programme, selon les besoins de l'utilisateur. Pour ****
*** soumettre une macro, il suffit de spécifier un énoncé indiquant le nom de ****
*** la macro à soumettre et les paramètres à utiliser. Dans ce programme, les ****
*** paramètres indiquent quel fichier doit être lu, quelles variables seront ****
*** utilisées, combien de poids bootstrap seront utilisés, et combien ****
*** d'échantillons de bootstrap ont été choisis. Le nom du fichier à lire est ****
*** toujours le même (BS_NPHS) et les deux derniers paramètres ont déjà été ****
*** définis au début du programme. L'utilisateur n'a donc qu'à spécifier les ****
*** variables à analyser. ****
***
*** Chaque soumission d'une macro donnera une estimation de la variance d'un ****
*** seul paramètre à estimer. Si plus d'un paramètre et sa variance doivent ****
*** être calculés, il faut alors soumettre la macro à plusieurs reprises. ****
***
*** Un exemple d'un énoncé pour soumettre une macro apparaît dans le programme ****
*** à l'endroit où l'utilisateur doit spécifier cet énoncé. L'exemple est mis ****
*** en commentaire dans le programme. L'énoncé utilisé par l'utilisateur sera ****
*** identique à l'exemple, à l'exception des variables à analyser. ****
*****/

```

```

/*****/
/* */
/* Total */
/* */
/*****/

proc datasets library=work; /* initialise alltots */
delete alltots;
run;

/*****/
*** Pour soumettre la macro total, l'énoncé suivant est utilisé: ***/
*** %total (infile, var, bssz=, multi=); ***/
*** où infile : bs_nphs ***/
*** var : la variable pour laquelle un total doit être ***/
*** calculé ***/
*** bssz= : bssz=&B. 00 ***/
*** multi= : multi=&R ***/
*** */
/*****/

* %total (bs_nphs, VAR, bssz=&B. 00, multi=&R);
%total (bs_nphs, di ab, bssz=&B. 00, multi=&R);
%total (bs_nphs, di ab2544, bssz=&B. 00, multi=&R);
%total (bs_nphs, mdi a45, bssz=&B. 00, multi=&R);
%total (bs_nphs, fdi a45, bssz=&B. 00, multi=&R);

/* Détruit les fichiers SAS de la macro total */

proc datasets library=work;
delete ytot est;
run;

/*****/
/* */
/* Rapport */
/* */
/*****/

/*****/
*** Pour soumettre la macro ratio, l'énoncé suivant est utilisé: ***/
*** %ratio (infile, var1, var2, bssz=, multi=); ***/
*** où infile : bs_nphs ***/
*** var1 : la variable au numérateur du rapport ***/
*** var2 : la variable au dénominateur du rapport ***/
*** bssz= : bssz=&B. 00 ***/
*** multi= : multi=&R ***/
*** */
/*****/

* %ratio (bs_nphs, VAR1, VAR2, bssz=&B. 00, multi=&R);
%ratio (bs_nphs, di ab, total, bssz=&B. 00, multi=&R);
%ratio (bs_nphs, mdi a45, mage45, bssz=&B. 00, multi=&R);
%ratio (bs_nphs, di ab45, age45, bssz=&B. 00, multi=&R);

/* Détruit les fichiers SAS de la macro ratio */

proc datasets library=work;
delete ytot xtot est;
run;

&by proc sort data=alltots;
&by by region;
&by run;

/*****/
*** Les résultats des macros total et ratio peuvent ***/
*** être sauvegardés dans un fichier permanent ***/
/*****/

data in3.ZZZ;
set alltots;
run;

%prnttot;

.
.
.

data time;
set time;
format stop datetime16.;
stop=datetime();
output;
run;

proc print data=time;
title 'Temps requis pour exécuter le programme';
run;

```

```
/* Fin du programme SAS BOOTVARF */
```

(L'exécution de ce programme a pris environ 6 minutes.)

Exemple 2B: Régression logistique

```
.
.
.

/**** Section 1: Déclaration des variables macros ****/
/**** Section 1: Déclaration des variables macros ****/

%let R=01; /* paramètre du bootstrap moyen (fixe) */

/**** L'UTILISATEUR DOIT INDiquer LE NOMBRE DE POIDS BOOTSTRAP À UTILISER, ****/
/**** DIVISÉ PAR 100 ****/
/**** %let B=5; pour les fichiers de poids bootstrap BD5H35 et BD5H356 ****/
/**** ****/

%let B=5; /* nombre de poids de bootstrap Bx100 */

/**** L'UTILISATEUR DOIT INDiquer LE RÉPERTOIRE CONTENANT LE FICHIER ****/
/**** D'ANALYSE (in1) ET LE FICHIER CONTENANT LES POIDS DE BOOTSTRAP ****/
/**** (in2). LES FICHIERs DE SORTIES SERONT SAUVEGARDÉS DANS LE ****/
/**** RÉPERTOIRE SPÉCIFIÉ POUR in3. ****/
/**** ****/

libname in1 "c:\data";
libname in2 "c:\bootstrp";
libname in3 "c:\output";

/**** L'UTILISATEUR DOIT INDiquer LE NOM DU FICHIER SAS CONTENANT LES VARIABLES ****/
/**** D'ANALYSE ****/
/**** %let Mfile=xxx; ****/
/**** ****/

%let Mfile=in1.dumyh356; /* Nom du fichier SAS principal */

/**** L'UTILISATEUR DOIT INDiquer LE NOM DU FICHIER SAS CONTENANT LES POIDS DE ****/
/**** BOOTSTRAP. SI LE FICHIER SAS N'EXISTE PAS, IL DOIT ÊTRE CRÉÉ. ****/
/**** %let bsamp=bd5h35; (à utiliser avec le fichier DUMYH35.TXT) OU ****/
/**** %let bsamp=bd5h356; (à utiliser avec le fichier DUMYH356.TXT) ****/
/**** ****/

data in2.bd5h356;
  %let datafid="c:\bootstrp\bd5h356.txt";
  %include "c:\bootstrp\boot_in.sas";
run;

%let bsamp=in2.bd5h356; /* fichier SAS des poids bootstrap */

/**** L'UTILISATEUR DOIT SPÉCIFIER LA VALEUR DES PARAMÈTRES by, ****/
/**** wo ET kp. CEUX-CI INDiquENT SI L'ANALYSE EST FAITE AU ****/
/**** NIVEAU RÉGIONAL OU POUR L'ENSEMBLE DU FICHIER. ****/
/**** ****/
/**** SI L'ANALYSE EST FAITE POUR L'ENSEMBLE DU FICHIER, ALORS ****/
/**** %let by=*; ****/
/**** %let wo=*; ****/
/**** %let kp=*; ****/
/**** ****/
/**** SINON, SI L'ANALYSE EST FAITE AU NIVEAU RÉGIONAL, ALORS ****/
/**** %let by=*; ****/
/**** %let wo=*; ****/
/**** %let kp=region; ****/
/**** ****/

%let by=*;
%let wo=*;
%let kp=*;

data time;
  format start datetime16;
```

```

start=datetime();
output;
run;

/*****
*** Section 2: Déclaration des macros ****
*****/

.
.
.

*****,
* Section 3: Programme principal *,
*****,

/* Lit le fichier principal */

/*****
*** À CAUSE DU GRAND NOMBRE D'OBSERVATIONS, SEULES LES VARIABLES ****
*** REQUISES POUR FAIRE LES ANALYSES ET LES VARIABLES, ****
*** "REALUKEY" ET "PERSONID" DEVRAIENT ÊTRE CONSERVÉES DÉSORMAIS. ****
*****/

data nphs (index=(id=(realukey personid)));
set &MFile (keep= realukey personid ccc4_1v age1820 age2124 age2534
age3544 age4554 age5564 age6574 age7584 age85 femmes);
run;

/* Lit les poids de bootstrap */
/* FWGT est le même poids que WT54 ou WT64, selon le fichier analysé */

data bsamp (index=(id=(realukey personid)));
set &bsamp;
keep fwgt realukey personid bsw1-bsw&B.00;
run;

/* Apparie le fichier principal et les poids bootstrap */

&by data bs_nphs (index=(region));
&wo data bs_nphs;
merge nphs (in=in1) bsamp (in=in2);
by realukey personid;
if in1;
run;

proc datasets library=work;
delete nphs bsamp;
run;

/*****
*** Section 4: Énoncés pour soumettre les macros ****
*****/

.
.
.

/*****
/* ****
/* Régression Logistique */
/* ****
*****/

proc datasets library=work; /* initialise bs_reglg */
delete bs_reglg;
run;

```

```

/***** Pour soumettre la macro logreg, l'énoncé suivant est utilisé: *****/
/**** %logreg(infile, yvar, xvar, bssz=&B.00, multi=); *****/
/**** ou      infile : bs_nphs *****/
/****          yvar  : la variable dépendante *****/
/****          xvar  : les variables indépendantes. Les variables doivent ****/
/****                  être inscrites comme elles le seraient dans l'énon- ****/
/****                  cé d'un modèle (sans virgules entre elles). *****/
/****          bssz= : bssz=&B.00 *****/
/****          multi= : multi=&R *****/
/**** *****/
/**** *****/

title "Estimation de la variance à l'aide du bootstrap &B.00: &R pour des";
title2 "Régressions logistiques";

* %logreg(bs_nphs, YVAR, XVAR, bssz=&B.00, multi=&R);
%logreg(bs_nphs, ccc4_iv, age1820 age2124 age2534 age3544
        age4554 age5564 age6574 age7584 age85 femmes, bssz=&B.00, multi=&R);

/* Détruit les fichiers SAS de la macro logreg */
proc datasets library=work;
    delete betas betat bsbeta origest;
run;

.
.
.

%prntlog;

data time;
    set time;
    format stop datetime16.;
    stop=datetime();
    output;
run;

proc print data=time;
    title 'Temps requis pour exécuter le programme';
run;

/* Fin du programme SAS BOOTVARF */

```

(L'exécution de ce programme a pris environ 12 minutes.)

Les exemples suivant présentent les résultats des analyses réalisées à l'aide des programmes de l'exemple 2. L'exemple 3A présente les résultats des totaux et des rapports. Par exemple, si on s'intéresse au rapport du nombre d'hommes diabétiques âgés de 45 ans et plus au nombre total d'hommes âgés de 45 ans et plus, en Ontario, on se réfère à l'observation 41. La région 35 correspond à la province de l'Ontario (voir le dictionnaire de données inclus sur le CD-ROM pour connaître les codes associés à chacune des provinces) et la variable Type indique le type d'analyse, dans ce cas-ci, un rapport. On retrouve au numérateur du rapport la variable *mdia45* (VAR1) et au dénominateur la variable *mage45* (VAR2). L'estimation du rapport est de 6,79 (YHAT) avec un écart-type de 0,59 (BS_SD) et un coefficient de variation de 8,68 (BS_CV). L'intervalle de confiance à 95% pour cette estimation est (5,63% , 7,94%) (CIL95, CIU95). À la suite des résultats, on peut voir que l'exécution de ce programme a débuté à 15 heures et s'est terminé à 15 heures 18 minutes.

Les résultats de la régression logistique sont présentés à l'exemple 3B. Par exemple, le paramètre estimé pour la variable AGE85 est 1,98725 (BHAT) et le rapport de cotes est de 7,29543 (ODDS). La statistique de Wald et son seuil p associé pour ce paramètre sont de 17,912 (WALD) et p=0,00002 (PVALUE) respectivement. L'estimation de la variance et de l'écart-type

du paramètre estimé sont de 0,22048 (BS_VAR) et 0,46955 (BS_SD) et le coefficient de variation est de 23,63 (BS_CV). Finalement, l'intervalle de confiance pour le rapport de cotes est (2,90642 , 18,3123) (CIL95, CIU95). L'exécution de ce programme a débuté à 16 heures 28 minutes et s'est terminé à 16 heures 34 minutes.

Exemple 3: Résultats du programme BOOTVARF.SAS et temps requis pour l'exécution du programme

Exemple 3A: Totaux et rapports

Estimation de la variance à l'aide du bootstrap 500:01 pour des Totaux et des Rapports									
OBS	REGION	TYPE	VAR1	VAR2	YHAT	BS_SD	BS_CV	CIL95	CIU95
1	10	Total	di ab	Aucune	18543.63	1755.46	9.47	15102.93	21984.32
2	10	Total	di ab2544	Aucune	3359.67	919.84	27.38	1556.78	5162.56
3	10	Total	ndi a45	Aucune	4847.84	965.18	19.91	2956.08	6739.59
4	10	Total	fdi a45	Aucune	9797.05	1327.86	13.55	7194.46	12399.65
5	10	Rapport	di ab	total	3.84	0.36	9.47	3.12	4.55
6	10	Rapport	ndi a45	mage45	5.86	1.17	19.91	3.57	8.15
7	10	Rapport	di ab45	age45	8.71	1.01	11.54	6.74	10.68
8	11	Total	di ab	Aucune	4333.79	476.68	11.00	3399.49	5268.08
9	11	Total	di ab2544	Aucune	556.99	145.58	26.14	271.66	842.33
10	11	Total	ndi a45	Aucune	1843.26	296.01	16.06	1263.08	2423.45
11	11	Total	fdi a45	Aucune	1760.49	306.24	17.39	1160.27	2360.72
12	11	Rapport	di ab	total	3.96	0.44	11.00	3.10	4.81
13	11	Rapport	ndi a45	mage45	9.04	1.45	16.06	6.20	11.89
14	11	Rapport	di ab45	age45	8.37	0.99	11.84	6.43	10.31
15	12	Total	di ab	Aucune	32067.97	4020.60	12.54	24187.59	39948.34
16	12	Total	di ab2544	Aucune	8657.34	2420.88	27.96	3912.42	13402.25
17	12	Total	ndi a45	Aucune	13014.68	1941.95	14.92	9208.46	16820.90
18	12	Total	fdi a45	Aucune	9824.63	2055.09	20.92	5796.64	13852.62
19	12	Rapport	di ab	total	4.22	0.53	12.60	3.18	5.26
20	12	Rapport	ndi a45	mage45	9.16	1.37	14.92	6.48	11.85
21	12	Rapport	di ab45	age45	7.56	1.03	13.63	5.54	9.58
22	13	Total	di ab	Aucune	20874.93	2386.72	11.43	16196.96	25552.90
23	13	Total	di ab2544	Aucune	2942.62	795.13	27.02	1384.16	4501.08
24	13	Total	ndi a45	Aucune	7346.71	1255.13	17.08	4886.65	9806.77
25	13	Total	fdi a45	Aucune	9372.10	1703.40	18.18	6033.43	12710.78
26	13	Rapport	di ab	total	3.33	0.38	11.43	2.59	4.08
27	13	Rapport	ndi a45	mage45	6.44	1.10	17.08	4.29	8.60
28	13	Rapport	di ab45	age45	6.95	0.93	13.32	5.14	8.77
29	24	Total	di ab	Aucune	180260.63	16620.08	9.22	147685.28	212835.99
30	24	Total	di ab2544	Aucune	28470.25	4499.07	15.80	19652.06	37288.43
31	24	Total	ndi a45	Aucune	74393.74	10925.90	14.69	52978.99	95808.50
32	24	Total	fdi a45	Aucune	73430.13	9766.00	13.30	54288.77	92571.49
33	24	Rapport	di ab	total	3.00	0.28	9.23	2.45	3.54
34	24	Rapport	ndi a45	mage45	6.77	0.99	14.67	4.82	8.71
35	24	Rapport	di ab45	age45	6.27	0.68	10.77	4.94	7.59
36	35	Total	di ab	Aucune	289230.27	15856.72	5.48	258151.10	320309.44
37	35	Total	di ab2544	Aucune	52825.50	8070.87	15.28	37006.60	68644.40
38	35	Total	ndi a45	Aucune	111913.46	9723.20	8.69	92855.99	130970.93
39	35	Total	fdi a45	Aucune	119382.33	9525.09	7.98	100713.16	138051.50
40	35	Rapport	di ab	total	3.20	0.18	5.48	2.85	3.54
41	35	Rapport	ndi a45	mage45	6.79	0.59	8.68	5.63	7.94
42	35	Rapport	di ab45	age45	6.63	0.38	5.75	5.88	7.38
43	46	Total	di ab	Aucune	29218.29	3292.51	11.27	22764.97	35671.62
44	46	Total	di ab2544	Aucune	2022.59	821.40	40.61	412.65	3632.53
45	46	Total	ndi a45	Aucune	12117.86	1992.33	16.44	8212.90	16022.83
46	46	Total	fdi a45	Aucune	14762.89	2737.45	18.54	9397.48	20128.29
47	46	Rapport	di ab	total	3.28	0.37	11.27	2.56	4.01
48	46	Rapport	ndi a45	mage45	7.27	1.20	16.44	4.93	9.62
49	46	Rapport	di ab45	age45	7.58	0.88	11.60	5.86	9.31
50	47	Total	di ab	Aucune	26075.51	2853.72	10.94	20482.21	31668.80
51	47	Total	di ab2544	Aucune	4374.16	1115.76	25.51	2187.27	6561.06
52	47	Total	ndi a45	Aucune	11237.90	1834.82	16.33	7641.66	14834.15
53	47	Total	fdi a45	Aucune	10041.21	1639.64	16.33	6827.52	13254.90
54	47	Rapport	di ab	total	3.30	0.36	10.95	2.59	4.01
55	47	Rapport	ndi a45	mage45	7.40	1.21	16.33	5.03	9.76
56	47	Rapport	di ab45	age45	6.71	0.81	12.06	5.12	8.29
57	48	Total	di ab	Aucune	56223.35	6812.45	12.12	42870.95	69575.76
58	48	Total	di ab2544	Aucune	11491.94	2711.18	23.59	6178.03	16805.85
59	48	Total	ndi a45	Aucune	26367.09	4722.15	17.91	17111.68	35622.49
60	48	Total	fdi a45	Aucune	16376.70	3680.89	22.48	9162.15	23591.26
61	48	Rapport	di ab	total	2.60	0.31	12.12	1.98	3.21
62	48	Rapport	ndi a45	mage45	7.33	1.31	17.91	4.76	9.90
63	48	Rapport	di ab45	age45	5.80	0.82	14.19	4.18	7.41
64	59	Total	di ab	Aucune	78631.91	8542.75	10.86	61888.13	95375.69
65	59	Total	di ab2544	Aucune	10884.13	3162.82	29.06	4684.99	17083.26
66	59	Total	ndi a45	Aucune	38529.16	5927.52	15.38	26911.22	50147.09
67	59	Total	fdi a45	Aucune	28314.56	5199.84	18.36	18122.87	38506.25
68	59	Rapport	di ab	total	2.59	0.28	10.86	2.04	3.14
69	59	Rapport	ndi a45	mage45	6.67	1.03	15.38	4.66	8.68
70	59	Rapport	di ab45	age45	5.59	0.66	11.76	4.30	6.88

Temps requis pour executer le programme

OBS START STOP
1 31JUL01: 15: 00: 15 31JUL01: 15: 18: 27

Exemple 3B: Régression logistique

Estimation de la variance à l'aide du bootstrap 500: 01 pour des
Régressions logistiques
Variable dépendante: ccc4_1v

OBS	BETA	BHAT	ODDS	WALD	PVALUE	BS_VAR	BS_SD	BS_CV	CI195	CIU95
1	Intercep	-3. 73740	0. 02382	287. 672	0. 00000	0. 04856	0. 22035	5. 90	0. 01546	0. 0367
2	age1820	0. 44095	1. 55418	1. 773	0. 18301	0. 10967	0. 33116	75. 10	0. 81211	2. 9743
3	age2124	0. 65935	1. 93353	4. 786	0. 02869	0. 09083	0. 30138	45. 71	1. 07106	3. 4905
4	age2534	0. 94600	2. 57539	16. 338	0. 00005	0. 05477	0. 23404	24. 74	1. 62789	4. 0744
5	age3544	1. 05200	2. 86338	20. 189	0. 00001	0. 05482	0. 23413	22. 26	1. 80959	4. 5308
6	age4554	1. 22279	3. 39664	25. 632	0. 00000	0. 05833	0. 24152	19. 75	2. 11574	5. 4530
7	age5564	1. 25142	3. 49531	24. 967	0. 00000	0. 06273	0. 25045	20. 01	2. 13943	5. 7105
8	age6574	1. 15386	3. 17042	22. 055	0. 00000	0. 06037	0. 24570	21. 29	1. 95872	5. 1317
9	age7584	1. 28205	3. 60402	25. 031	0. 00000	0. 06566	0. 25625	19. 99	2. 18104	5. 9554
10	age85	1. 98725	7. 29543	17. 912	0. 00002	0. 22048	0. 46955	23. 63	2. 90642	18. 3123
11	femmes	0. 21987	1. 24592	7. 535	0. 00605	0. 00642	0. 08010	36. 43	1. 06489	1. 4577

Temps requis pour executer le programme

OBS START STOP
1 30JUL01: 16: 28: 00 30JUL01: 16: 34: 41

3.3 Modifications au programme aux fins de test

Comme il a été mentionné précédemment, l'exécution du programme peut prendre un certain temps dans le cas des analyses de régression. Il est possible de réduire le nombre de poids bootstrap utilisés afin de *tester* le programme (**Cependant, pour obtenir les estimations de variance, il est indispensable d'utiliser l'ensemble complet des poids bootstrap fournis**). Par exemple, dans le cas d'une régression logistique, il suffit de spécifier le nombre de fois où la régression sera calculée à l'aide de poids bootstrap (par exemple, 2 fois au lieu de 500 fois) dans la routine "logreg" à la section 2 du programme. L'utilisateur doit ajouter trois énoncés comme montré en *italiques* dans l'exemple 4. La valeur de kL représente le nombre de fois où la régression sera calculée (dans ce cas-ci, 2 fois). Les deux autres paramètres, k et j, doivent égaier 10 et 1 respectivement pour le test. À noter que ces modifications s'ajoutent aux paramètres à spécifier comme montré dans l'exemple 2B. Encore une fois, l'estimation de la variance calculée ainsi ne sera pas valide.

On doit également spécifier le nombre de poids bootstrap à lire à la section 3. L'exemple suivant indique où le changement doit être fait. La partie en ***italique*** doit remplacer la partie entre parenthèses. Le même changement peut être fait pour les routines "regress" et "regglm".

Exemple 4: Changement au nombre d'itération aux fins de test

```

.
.
.

/***** Section 2: Déclaration des macros *****/
*****/

.
.
.

*****,

%macro logreg(infile, yvar, xvar, bssz=, multi=);

*****,
proc logistic data=infile outest=orig(keep=&kp intercep &xvar) descending noprint;
  model &yvar=&xvar;
  &by by region;
  weight fwgt;
run;

proc transpose data=orig out=origest prefix=bhat name=beta;
  var intercep &xvar;
  &by by region;
run;

%let L=%eval (&bssz/10);

%do k=1 %to 10;
  %let j=%eval (1+((&k-1)*&L));
  %let kL=%eval (&k*&L);

  %let k=10;
  %let j=1;
  %let kL=2;

  data poi ds;
    set &infile;
    keep bsw&j- bsw&kL &yvar &xvar &kp;
  run;

  %do i=&j %to &kL;

    proc logistic data=poi ds outest=betas(keep=&kp intercep &xvar) noprint descending;
      model &yvar=&xvar;
      &by by region;
      weight bsw&i;
    run;

    proc transpose data=betas out=betat prefix=best name=beta;
      var intercep &xvar;
      &by by region;
    run;

    data betat;
      set betat;
      rename best1=best&i;
    run;

    %if (&i =1) %then %do;

      data bsbeta;
        set betat;
      run;

    %end;
    %else %do;

      data bsbeta;
        merge bsbeta betat;
        &by by region;
      run;

    %end;
  %end;
%end;

data bs_regl;
  merge origest bsbeta;
  rename bhat1=bhat;
  bs_var=&multi *((&bssz-1)*(var(of best1-best&bssz)))/&bssz;
  bs_sd=sqrt(bs_var);
  bs_cv=abs(round((bs_sd/bhat1)*100,.01));
  wald=(bhat1/bs_sd)*(bhat1/bs_sd);
  pvalue=1-probchi(wald,1);

```

```

lo95=bhat1-1.96*bs_sd;
hi95=bhat1+1.96*bs_sd;
odds=exp(bhat1);
ci_l95=exp(lo95);
ci_u95=exp(hi95);
ydep="&yvar";
drop best1-best&bssz;
run;

%let printlog=1;
%let dep=&yvar;

%mend logreg;

.
.
.

*****
* Section 3: Programme principal
*****

/* Lit le fichier principal */

/*****
*** À CAUSE DU GRAND NOMBRE D'OBSERVATIONS, SEULES LES VARIABLES ***
*** REQUISES POUR FAIRE LES ANALYSES ET LES VARIABLES. ***
*** "REALUKEY" ET "PERSONID" DEVRAIENT ÊTRE CONSERVÉES DÉSORMAIS. ***
*****/

data nphs (index=(id=(realukey personid)));
  set &Mfile (keep= realukey personid ccc4_v age1820 age2124 age2534
                 age3544 age4554 age5564 age6574 age7584 age85 femmes);
run;

/* Lit les poids de bootstrap */
/* FWGT est le même poids que WT54 ou WT64, selon le fichier analysé */

data bsamp (index=(id=(realukey personid)));
  set &bsamp;
  keep fwgt realukey personid bsw1-bsw2; (au lieu de bsw1-bsw&B.00;)
run;

```

(Le programme de l'exemple 2B a été testé de cette façon et l'exécution a pris environ 2 minutes.)

Programme BOOTVARF.SAS

*

AVERTISSEMENT

* Le gouvernement du Canada (Statistique Canada) possède tous les droits de propriété intellectuelle (y compris le droit d'auteur) relatifs à ce logiciel. Sous réserve des modalités décrites ci-dessous, vous obtenez une licence non exclusive et non transmissible d'utilisation de ce logiciel.

* Ce logiciel vous est fourni « tel quel », et le titulaire ne donne aucune garantie expresse ou implicite, ce qui comprend les garanties de qualité marchande et de convenance à une fin particulière mais ne se limite pas à ces garanties. En aucune circonstance le titulaire ne sera responsable de dommages indirects, particuliers ou consécutifs, ou d'autres dommages semblables. Ce contrat sera automatiquement résilié, sans préavis, si vous omettez de respecter une quelconque des modalités du contrat.;

```

/*****
/* Date: 1er décembre 1997
/* Modifié: 30 juillet 2001
/*****/

/*****/
/****
/****                               Programme SAS BOOTVAR
/****
/**** Ce programme calcule des estimations de variance pour différents types
/**** d'estimations à l'aide des poids bootstrap. À l'aide de macros SAS,
/**** ce programme peut calculer des estimations de variance pour des totaux,
/**** des rapports, et des différences de rapports. Il peut aussi calculer des
/**** estimations de variance pour des paramètres de régressions linéaires,
/**** logistiques ou d'un modèle linéaire généralisé. Ce programme peut aussi
/**** être modifié pour s'adapter à d'autres types d'analyse.
/****
/**** Ce programme est divisé en 4 sections, décrites plus bas. Tout au long
/**** du programme, les changements que l'utilisateur doit apporter sont indi-
/**** qués.
/****
/**** Section 1: Déclaration des variables macros
/**** C'est à cet endroit qu'on définit les variables qui seront
/**** utilisées tout au long du programme. Certains changements
/**** doivent être apportés par l'utilisateur.
/**** Section 2: Déclaration des macros
/**** C'est à cet endroit qu'est définie la portion du programme qui
/**** calcule les estimations de variance. Aucun changement n'est
/**** requis de la part de l'utilisateur dans cette section (à l'ex-
/**** ception de la macro pour un modèle linéaire généralisé (REGGLM).
/**** Cette macro doit être modifiée selon les besoins de
/**** l'utilisateur).
/**** Section 3: Programme principal
/**** C'est à cet endroit qu'on lit les fichiers contenant les
/**** variables à analyser et les poids bootstrap. Les variables
/**** à utiliser dans les analyses doivent être définies par l'utili-
/**** sateur avant d'utiliser ce programme.
/**** Section 4: Énoncés pour soumettre les macros
/**** C'est à cet endroit que l'utilisateur doit indiquer quelle(s)
/**** macro(s) il/elle désire soumettre. À ce stade, l'utilisateur
/**** peut soumettre une macro une fois, une macro plusieurs fois,
/**** ou plusieurs macros. Étant donné le grand nombre d'observa-
/**** tions à analyser, il est recommandé de limiter le
/**** nombre de macros qui sont soumises, en particulier dans le cas
/**** de régressions.
/****
/**** Avant d'utiliser ce programme, l'utilisateur doit préparer un fichier
/**** d'analyse qui contiendra les variables à analyser. Ce fichier ne devrait
/**** contenir que les enregistrements pour lesquels l'analyse est nécessaire
/**** (p.e. si l'analyse est faite pour le groupe d'âge 12 ans et plus, seuls
/**** les enregistrements des individus de 12 ans et plus devraient être inclus
/**** dans le fichier). Une variable appelée REGION doit aussi être créée si
/**** les analyses sont faites à un niveau régional. Par exemple, si l'analyse
/**** est faite au niveau provincial, la variable REGION prendra la même valeur
/**** que la variable qui contient la province sur le fichier de microdonnées
/**** (p.e. PRC4_CUR). Si l'analyse est faite au niveau des régions de santé,
/**** la variable REGION prendra la valeur de la région de santé pour chaque
/**** enregistrement.
/****
/**** N.B. Pour calculer un total, un rapport, ou une différence entre deux
/**** rapports, les estimations sont obtenues en sommant le poids des enregis-
/**** trements qui ont la caractéristique d'intérêt. Donc, une variable indi-
/**** catrice doit être créée pour chaque variable à analyser. Elle prendra la
/**** valeur 1 lorsque l'enregistrement a la caractéristique d'intérêt et 0
/**** sinon.
/*****/
```

options ps=48 ls=120 ;

```

/***** Section 1: Déclaration des variables macros *****/
/*****

%let R=01;                               /* paramètre du bootstrap moyen (fixe) */

/*****
/**** L'UTILISATEUR DOIT INDiquer LE NOMBRE DE POIDS BOOTSTRAP À UTILISER, ****/
/**** DIVISÉ PAR 100 ****/
/**** %let B=5; pour les fichiers de poids bootstrap BD5H35 et BD5H356 ****/
/****

%let B=5;                               /* nombre de poids Bootstrap Bx100 */

/*****
/**** L'UTILISATEUR DOIT INDiquer LE RÉPERTOIRE CONTENANT LE FICHIER ****/
/**** D' ANALYSE (in1) ET LE FICHIER CONTENANT LES POIDS BOOTSTRAP ****/
/**** (in2). LES FICHIERS DE SORTIES SERONT SAUVEGARDÉS DANS LE ****/
/**** RÉPERTOIRE SPÉCIFIÉ POUR in3. ****/
/****

libname in1 "c:\data";
libname in2 "c:\bootstrp";
libname in3 "c:\output";

/*****
/**** L'UTILISATEUR DOIT INDiquer LE NOM DU FICHIER SAS CONTENANT LES VARIABLES ****/
/**** D' ANALYSE. ****/
/**** %let Mfile=xxx; ****/
/****

%let Mfile=in1.dumyh356;                 /* Nom du fichier SAS principal */

/*****
/**** L'UTILISATEUR DOIT INDiquer LE NOM DU FICHIER TEXTE CONTENANT LES POIDS ****/
/**** BOOTSTRAP DE MÊME QUE LE FICHIER DU CLICHÉ D'ARTICLE. ****/
/**** bd5h35.txt et b35_i.sas; (à utiliser avec le fichier DUMYH35.TXT) OU ****/
/**** bd5h356.txt et b356_i.sas; (à utiliser avec le fichier DUMYH356.TXT) ****/
/****

data in2.bd5h356;
  %let datafid="c:\bootstrp\bd5h356.txt";
  %include "c:\bootstrp\boot_in.sas";
run;

%let bsamp=in2.bd5h356;                 /* fichier SAS des poids bootstrap */

/*****
/**** L'UTILISATEUR DOIT SPÉCIFIER LA VALEUR DES PARAMÈTRES by, ****/
/**** wo ET kp. CEUX-CI INDiquENT SI L'ANALYSE EST FAITE AU ****/
/**** NIVEAU RÉGIONAL OU POUR L'ENSEMBLE DU FICHIER. ****/
/****
/**** SI L'ANALYSE EST FAITE POUR L'ENSEMBLE DU FICHIER, ALORS ****/
/**** %let by=*, ****/
/**** %let wo=*, ****/
/**** %let kp=*, ****/
/****
/**** SINON, SI L'ANALYSE EST FAITE AU NIVEAU RÉGIONAL, ALORS ****/
/**** %let by=*, ****/
/**** %let wo=*, ****/
/**** %let kp=region; ****/
/****

%let by=*,
%let wo=*,
%let kp=;

data time;
  format start datetime16.;
  start=datetime();
  output;
run;

/*****
/**** Section 2: Déclaration des macros *****/
/****

%let printtot=0;
%let printdif=0;
%let printreg=0;
%let printlog=0;

%global dep;

```

```

*****;

%macro total(infile, var, bssz=, multi=);
*****;

proc means data=&infile noprint;
  var fwt bsw1- bsw&bssz;
  weight &var;
  &by by region;
  output out=ytot
    sum=yhat ybs1-ybs&bssz;
run;

data est;
  set ytot;
  length var1 $ 8;
  length var2 $ 8;
  length type $ 8;
  bs_var=&multi*((&bssz-1)*(var(of ybs1-ybs&bssz)))/&bssz;
  bs_sd=sqrt(bs_var);
  bs_cv=round((bs_sd/yhat)*100,.01);
  cil95=yhat-1.96*bs_sd;
  ciu95=yhat+1.96*bs_sd;
  var1="&var";
  var2="Aucune";
  type="Total";
  drop ybs1-ybs&bssz _type_ _freq_;
run;

proc append data=est base=alltots;
run;

%let printtot=1;

%mend total;

*****;

%macro ratio(infile, var1, var2, bssz=, multi=);
*****;

proc means data=&infile noprint;
  var fwt bsw1- bsw&bssz;
  weight &var1;
  &by by region;
  output out=ytot
    sum=yhat ybs1-ybs&bssz;
run;

proc means data=&infile noprint;
  var fwt bsw1- bsw&bssz;
  weight &var2;
  &by by region;
  output out=xtot
    sum=xhat xbs1-xbs&bssz;
run;

data est;
  merge ytot xtot;
  array ybs{&bssz};
  array xbs{&bssz};
  array rbs{&bssz};
  length var1 $ 8;
  length var2 $ 8;
  length type $ 8;
  yhat=((yhat/xhat)*100);
  do i=1 to &bssz;
    rbs{i}=((ybs{i}/xbs{i})*100);
  end;
  bs_var=&multi*((&bssz-1)*(var(of rbs1-rbs&bssz)))/&bssz;
  bs_sd=sqrt(bs_var);
  bs_cv=round((bs_sd/yhat)*100,.01);
  cil95=yhat-1.96*bs_sd;
  ciu95=yhat+1.96*bs_sd;
  var1="&var1";
  var2="&var2";
  type="Rapport";
  drop ybs1-ybs&bssz xbs1-xbs&bssz rbs1-rbs&bssz xhat i _type_ _freq_;
run;

proc append data=est base=alltots;
run;

%let printtot=1;

%mend ratio;

```

```

*****;

%macro diff_rat(infile, var1, var2, var3, var4, bssz=, multi=);
*****;

proc means data=&infile noprint;
  var fwgt bsw1- bsw&bssz;
  weight &var1;
  &by by region;
  output out=ytot
         sum=yhat ybs1-ybs&bssz;
run;

proc means data=&infile noprint;
  var fwgt bsw1- bsw&bssz;
  weight &var2;
  &by by region;
  output out=xtot
         sum=xhat xbs1-xbs&bssz;
run;

proc means data=&infile noprint;
  var fwgt bsw1- bsw&bssz;
  weight &var3;
  &by by region;
  output out=yytot
         sum=yyhat yybs1-yybs&bssz;
run;

proc means data=&infile noprint;
  var fwgt bsw1- bsw&bssz;
  weight &var4;
  &by by region;
  output out=xxtot
         sum=xxhat xxbs1-xxbs&bssz;
run;

data est;
  merge ytot xtot yytot xxtot;
  array ybs{&bssz};
  array xbs{&bssz};
  array yybs{&bssz};
  array xxbs{&bssz};
  array drbs{&bssz};
  length var1 $ 8;
  length var2 $ 8;
  length var3 $ 8;
  length var4 $ 8;
  length type $ 10;
  yhat=((yhat/xhat) - (yyhat/xxhat))*100;
  do i=1 to &bssz;
    drbs{i}=(((ybs{i}/xbs{i}) - (yybs{i}/xxbs{i}))*100);
  end;
  bs_var=(&multi*((&bssz-1)*(var(of drbs1-drbs&bssz)))/&bssz);
  bs_sd=sqrt(bs_var);
  bs_cv=abs(round((bs_sd/yhat)*100,.01));
  cil95=yhat-1.96*bs_sd;
  ciu95=yhat+1.96*bs_sd;
  var1="&var1";
  var2="&var2";
  var3="&var3";
  var4="&var4";
  type="Diff_Rapport";
  drop ybs1-ybs&bssz xbs1-xbs&bssz yybs1-yybs&bssz xxbs1-xxbs&bssz drbs1-drbs&bssz
      xhat xxhat yyhat i _type_ _freq_;
run;

proc append data=est base=diff_rat;
run;

%let printdif=1;

%mend diff_rat;

```

```

*****;

%macro regress(infile, yvar, xvar, bssz=, multi=);
*****;

proc reg data=infile outest=orig(keep=&kp intercep &xvar) noprint;
  model &yvar=&xvar;
  weight fwt;
  &by by region;
run;

proc transpose data=orig out=origest(drop=_label_) prefix=bhat name=beta;
  var intercep &xvar;
  &by by region;
run;

%let L=%eval (&bssz/10);

%do k=1 %to 10;
  %let j=%eval (1+((&k-1)*&L));
  %let kL=%eval (&k*&L);

  data pois;
    set &infile;
    keep bsw&j- bsw&kL &yvar &xvar &kp;
  run;

  %do i=&j %to &kL;

    proc reg data=pois outest=betas(keep=&kp intercep &xvar) noprint;
      model &yvar=&xvar;
      weight bsw&i;
    &by by region;
  run;

    proc transpose data=betas out=betat prefix=best name=beta;
      var intercep &xvar;
    &by by region;
  run;

    data betat;
      set betat;
      drop _label_;
      rename best1=best&i;
    run;

    %if (&i =1) %then %do;

      data bsbeta;
        set betat;
      run;

    %end;
    %else %do;

      data bsbeta;
        merge bsbeta betat;
    &by by region;
  run;

    %end;
  %end;
%end;

data est;
  merge origest bsbeta;
  rename bhat1=bhat;
  bs_var=&multi*((&bssz-1)*(var(of best1-best&bssz)))/&bssz;
  bs_sd=sqrt(bs_var);
  bs_cv=abs(round((bs_sd/bhat1)*100,.01));
  cil95=bhat1-1.96*bs_sd;
  ciu95=bhat1+1.96*bs_sd;
  ydep="&yvar";
  drop best1-best&bssz;
run;

proc append data=est base=bs_reg;
run;

%let printreg=1;
%let dep=&yvar;

%mend regress;

```

```

*****;

%macro logreg(infile, yvar, xvar, bssz=, multi=);
*****;
proc logistic data=&infile outest=orig(keep=&kp intercept &xvar) descending noprint;
  model &yvar=&xvar;
  &by by region;
  weight fwgt;
run;

proc transpose data=orig out=origest prefix=bhat name=beta;
  var intercept &xvar;
  &by by region;
run;

%let L=%eval (&bssz/10);

%do k=1 %to 10;
  %let j=%eval (1+((&k-1)*&L));
  %let kL=%eval (&k*&L);

  data poi ds;
    set &infile;
    keep bsw&j- bsw&kL &yvar &xvar &kp;
  run;

  %do i=&j %to &kL;

    proc logistic data=poi ds outest=betas(keep=&kp intercept &xvar) noprint descending;
      model &yvar=&xvar;
      &by by region;
      weight bsw&i;
    run;

    proc transpose data=betas out=betat prefix=best name=beta;
      var intercept &xvar;
      &by by region;
    run;

    data betat;
      set betat;
      rename best1=best&i;
    run;

    %if (&i =1) %then %do;

      data bsbeta;
        set betat;
      run;

    %end;
    %else %do;

      data bsbeta;
        merge bsbeta betat;
        &by by region;
      run;

    %end;
  %end;
%end;

data est;
  merge origest bsbeta;
  rename bhat1=bhat;
  bs_var=&multi*((&bssz-1)*(var(of best1-best&bssz)))/&bssz;
  bs_sd=sqrt(bs_var);
  bs_cv=abs(round((bs_sd/bhat1)*100,.01));
  wald=(bhat1/bs_sd)*(bhat1/bs_sd);
  pvalue=1-probchi(wald,1);
  lo95=bhat1-1.96*bs_sd;
  hi95=bhat1+1.96*bs_sd;
  odds=exp(bhat1);
  ci195=exp(lo95);
  ciu95=exp(hi95);
  ydep="&yvar";
  drop best1-best&bssz;
run;

proc append data=est base=bs_reglg;
run;

%let printlog=1;
%let dep=&yvar;

%mend logreg;

```

```

*****;

%macro regglm(infile, yvar, xvar, bssz=, multi=);
*****;
proc glm data=infile noprint;
  class &xvar;
  model &yvar= &xvar;
  LSMEANS &xvar / out=orig;
  weight fwgt;
&by by region;
run;

data orig2;
  set orig;
  drop _name_ stderr;
  array meanss{3};
  do i=1 to 3;
    if _n_ = i then do;
      meanss(i)=lsmean;
      retain;
    end;
  end;

  if _n_=3 then do;
    di ff1_2= meanss1-meanss2;
    di ff1_3= meanss1-meanss3;
    di ff2_3= meanss2-meanss3;
    drop lsmean i meanss1- meanss3;
    output;
  end;
run;

proc transpose data=orig2 out=origest prefix=bhat name=beta;
var di ff1_2 di ff1_3 di ff2_3;
run;

%let L=%eval (&bssz/10);

%do k=1 %to 10;
  %let j=%eval (1+((&k-1)*&L));
  %let kL=%eval (&k*&L);

  data poid;
    set &infile;
    keep bsw&j- bsw&kL &yvar &xvar &kp;
  run;

  %do i=&j %to &kL;

  proc glm data=poid;
  class &xvar;
  model &yvar= &xvar;
  LSMEANS &xvar / out=betas;
  weight bsw&i;
  &by by region;
run;

data betas2;
  set betas;
  drop _name_ stderr;
  array meanss{3};
  do i=1 to 3;
    if _n_ = i then do;
      meanss(i)=lsmean;
      retain;
    end;
  end;

  if _n_=3 then do;
    di ff1_2= meanss1-meanss2;
    di ff1_3= meanss1-meanss3;
    di ff2_3= meanss2-meanss3;
    drop lsmean i meanss1- meanss3;
    output;
  end;
run;

proc transpose data=betas2 out=betat prefix=best name=beta;
var di ff1_2 di ff1_3 di ff2_3;
run;

data betat;
  set betat;
  rename best1=best&i;
run;

%if (&i =1) %then %do;

  data bsbeta;
    set betat;

```

```

run;

%end;
%else %do;

    data bsbeta;
        merge bsbeta betat;
        &by by region;
    run;

%end;
%end;
%end;

data est;
    merge origest bsbeta;
    rename bhat1=bhat;
    bs_var=&multi*((&bssz-1)*(var(of best1-best&bssz)))/&bssz;
    bs_sd=sqrt(bs_var);
    bs_cv=abs(round((bs_sd/bhat1)*100,.01));
    cil95=bhat1-2.39406*bs_sd;
    ciu95=bhat1+2.39406*bs_sd;
    drop best1-best&bssz;
run;

proc append data=est base=bs_reg;
run;

%let printreg=1;
%let dep=&yvar;

%mend regglm;

*****;

%macro prnttot;

*****;
%if &prnttot=1 %then %do;

&by proc sort data=alltots;
&by by region;
&by run;

/*****
/* Imprime les résultats des macro total et ratio */
*****/

proc print data=alltots;
    title "Estimation de la variance à l'aide du bootstrap &B.00: &R pour des";
    title2 "Totaux et des Rapports";
    var &kp type var1 var2 yhat bs_sd bs_cv cil95 ciu95;
    format yhat bs_sd cil95 ciu95 11.2;
run;
%end;

/*****
/* 00: ****/
/* type : type d'estimation (Total ou Rapport) ****/
/* var1 et var2 : variables utilisées pour calculer ****/
/* les estimations. Pour un total, ****/
/* var2=Aucune. ****/
/* yhat : estimation du paramètre (en % pour ****/
/* un rapport) ****/
/* bs_sd : écart-type ****/
/* bs_cv : coefficient de variation ****/
/* cil95 : borne inférieure de l'intervalle de ****/
/* confiance à 95% ****/
/* ciu95 : borne supérieure de l'intervalle de ****/
/* confiance à 95% ****/
*****/

%mend prnttot;

*****;

%macro prntdiff;

*****;
%if &prntdiff=1 %then %do;

&by proc sort data=diffrat;
&by by region;
&by run;

/*****
/* Imprime les résultats de la macro diff_rat */
*****/

```

```

proc print data=di ffrat;
  title "Estimation de la variance à l'aide du bootstrap &B. 00: &R pour des";
  title2 "Différences entre des Rapports";
  var &kp type var1 var2 var3 var4 yhat bs_sd bs_cv cil95 ciu95;
  format yhat bs_sd cil95 ciu95 11.2;
run;
%end;

/*****
*** Oû: ****
*** type : type d'estimation (Diff_Rapport) ****
*** var1, var2, ****
*** var3 et var4 : variables utilisées pour calculer ****
*** l'estimation ****
*** yhat : estimation (différence en %) ****
*** bs_sd : écart-type ****
*** bs_cv : coefficient de variation ****
*** cil95 : borne inférieure de l'intervalle de ****
*** confiance à 95% ****
*** ciu95 : borne supérieure de l'intervalle de ****
*** confiance à 95% ****
*****/

%mend prntdiff;

*****;

%macro prntreg;
*****;
%if &printreg=1 %then %do;

/*****
/* Imprime les résultats des macros regress et regglm */
*****/

proc print data=bs_reg;
  title "Estimation de la variance à l'aide du bootstrap &B. 00: &R pour une";
  title2 "Régression";
  title3 "Variable dépendante: &dep";
  var &kp beta bhat bs_var bs_sd bs_cv cil95 ciu95;
run;
%end;

/*****
*** Oû: ****
*** beta : paramètre à estimer ****
*** bhat : estimation du paramètre ****
*** bsvar : variance de l'estimation du paramètre */
*** bs_sd : écart-type de l'estimation du ****
*** paramètre ****
*** bs_cv : coefficient de variation de l'esti- ****
*** mation du paramètre ****
*** cil95 : borne inférieure de l'intervalle de ****
*** confiance à 95% ****
*** ciu95 : borne supérieure de l'intervalle de ****
*** confiance à 95% ****
*****/

%mend prntreg;

*****;

%macro prntlog;
*****;
%if &printlog=1 %then %do;

/*****
/* Imprime les résultats de la macro logreg */
*****/

proc print data=bs_reglg;
  title "Estimation de la variance à l'aide du bootstrap &B. 00: &R pour des";
  title2 "Régressions logistiques";
  title3 "Variable dépendante: &dep";
  var &kp beta bhat odds wald pvalue bs_var bs_sd bs_cv cil95 ciu95;
run;
%end;

```

```

/*****
*** Ou:
*** beta      : paramètre à estimer
*** bhat      : estimation du paramètre
*** odds      : rapport de cotes
*** wald      : statistique de Wald
*** pvalue    : valeur p de la statistique de Wald
*** bsvar     : variance de l'estimation du paramètre
*** bs_sd     : écart-type de l'estimation du paramètre
*** bs_cv     : coefficient de variation de l'estimation du paramètre
*** cil95     : borne inférieure de l'intervalle de confiance à 95% du rapport de cotes
*** ciu95     : borne supérieure de l'intervalle de confiance à 95% du rapport de cotes
*****/

%mend prntlog;

*****
* Section 3: Programme principal
*****

/* Lit le fichier principal */

/*****
*** À CAUSE DU GRAND NOMBRE D'OBSERVATIONS, SEULES LES VARIABLES
*** REQUISES POUR FAIRE LES ANALYSES ET LES VARIABLES
*** "REALUKEY" ET "PERSONID" DEVRAIENT ÊTRE CONSERVÉES DÉSORMAIS.
*****/

data nphs (index=(id=(realukey personid)));
set &Mfile (keep= realukey personid + VARIABLES À ANALYSER);
run;

/* Lit les poids bootstrap */
/* FWGT correspond à WT54 ou WT64, selon le fichier à analyser */

data bsamp (index=(id=(realukey personid)));
set &bsamp;
keep fwgt realukey personid bsw1-bsw&B. 00;
run;

/* Apparie le fichier principal et les poids bootstrap */

&by data bs_nphs (index=(region));
&wo data bs_nphs;
merge nphs (in=in1) bsamp (in=in2);
by realukey personid;
if in1;
run;

proc datasets library=work;
delete nphs bsamp;
run;

```



```

/*****
*** Les résultats des macro total et ratio ***
*** peuvent être sauvegardés dans un fichier permanent ***
*****/

* data in3.ZZZ;
* set alltots;
* run;

/*****
/*
/* Différence entre des Rapports
/*
*****/

proc datasets library=work; /* initialise diff_rat */
delete diff_rat;
run;

/*****
*** Pour soumettre la macro diff_rat, l'énoncé suivant est utilisé: ***
*** %diff_rat(infile, var1, var2, var3, var4, bssz=, multi=); ***
*** où infile : bs_nphs ***
*** var1 : la variable au numérateur du 1er rapport ***
*** var2 : la variable au dénominateur du 1er rapport ***
*** var3 : la variable au numérateur du 2e rapport ***
*** var4 : la variable au dénominateur du 2e rapport ***
*** bssz= : bssz=&B.00 ***
*** multi= : multi=&R ***
*** Nota: L'intervalle de confiance est calculé pour une analyse faisant ***
*** une seule comparaison de rapports. Si des comparaisons multiples sont ***
*** faites, la façon de calculer l'intervalle de confiance doit tenir ***
*** compte de cette situation. Pour cette raison, dans le cas de comparai- ***
*** sons multiples, la valeur de la loi normale (Z) utilisée dans le ***
*** calcul de l'intervalle de confiance doit être corrigée à la section 2, ***
*** Déclaration des macros, en utilisant, par exemple, l'approche de ***
*** Bonferroni pour les comparaisons multiples. ***
*****/

* %diff_rat(bs_nphs, VAR1, VAR2, VAR3, VAR4, bssz=&B.00, multi=&R);

/* Détruit les fichiers SAS de la macro diff_rat */
proc datasets library=work;
delete ytot xtot yttot xttot est;
run;

%prntdiff;

/*****
*** Les résultats de la macro diff_rat peuvent ***
*** être sauvegardés dans un fichier permanent ***
*****/

* data in3.ZZZ;
* set diff_rat;
* run;

/*****
/*
/* Régression
/*
*****/

proc datasets library=work; /* initialise bs_reg */
delete bs_reg;
run;

/*****
*** Pour soumettre la macro regress, l'énoncé suivant est utilisé: ***
*** %regress(infile, yvar, xvar, bssz=, multi=); ***
*** où infile : bs_nphs ***
*** yvar : la variable dépendante ***
*** xvar : les variables indépendantes. Les variables doivent ***
*** être inscrites comme elles le seraient dans l'énon- ***
*** cé d'un modèle (sans virgules entre elles). ***
*** bssz= : bssz=&B.00 ***
*** multi= : multi=&R ***
*****/

* %regress(bs_nphs, YVAR, XVAR, bssz=&B.00, multi=&R);

/* Détruit les fichiers SAS de la macro regress */
proc datasets library=work;
delete betas betat bsbeta origest;

```

```

run;

/*****
/*
/* Régression Logistique */
/*
/*
*****/

proc datasets library=work; /* initialise bs_reglg */
delete bs_reglg;
run;

/*****
*** Pour soumettre la macro logreg, l'énoncé suivant est utilisé: ***
*** %logreg(infile, yvar, xvar, bssz=, multi=); ***
*** où infile : bs_nphs ***
*** yvar : la variable dépendante ***
*** xvar : les variables indépendantes. Les variables doivent ***
*** être inscrites comme elles le seraient dans l'énon- ***
*** cé d'un modèle (sans virgules entre elles). ***
*** bssz= : bssz=&B.00 ***
*** multi= : multi=&R ***
*****/

* %logreg(bs_nphs, YVAR, XVAR, bssz=&B.00, multi=&R);

/* Détruit les fichiers SAS de la macro logreg */
proc datasets library=work;
delete betas betat bsbeta origest;
run;

/*****
/*
/* Modèle Linéaire Généralisé */
/*
/*
*****/

/*****
*** N.B. La macro regglm doit être modifiée pour répondre aux besoins de ***
*** l'utilisateur. La macro regglm définie dans ce programme est un ***
*** exemple d'une analyse qui a été faite en utilisant ce programme. ***
*****/
*** Pour soumettre la macro regglm, l'énoncé suivant est utilisé: ***
*** %regglm(infile, yvar, xvar, bssz=, multi=); ***
*** où infile : bs_nphs ***
*** yvar : la variable dépendante ***
*** xvar : les variables indépendantes. Les variables doivent ***
*** être inscrites comme elles le seraient dans l'énon- ***
*** cé d'un modèle (sans virgules entre elles). ***
*** bssz= : bssz=&B.00 ***
*** multi= : multi=&R ***
*****/
*** Nota: Si la macro regglm est adaptée pour calculer des différences entre ***
*** des moyennes, la façon de calculer l'intervalle de confiance doit ***
*** tenir compte des comparaisons multiples. La valeur de la loi ***
*** normale (Z) utilisée dans le calcul de l'intervalle de confiance ***
*** doit être corrigée à la section 2, Déclaration des macros, en ***
*** utilisant, par exemple, l'approche de Bonferroni pour les ***
*** comparaisons multiples. ***
*****/

* %regglm(bs_nphs, YVAR, XVAR, bssz=&B.00, multi=&R);

/* Détruit les fichiers SAS de la macro regglm */
proc datasets library=work;
delete betas betat bsbeta origest;
run;

%prntreg;

/*****
*** Les résultats des macros regress et regglm ***
*** peuvent être sauvegardés dans un fichier permanent ***
*****/

* data in3.ZZZ;
* set bs_reg;
* run;

%prntlog;

/*****
*** Les résultats de la macro logreg peuvent ***
*** être sauvegardés dans un fichier permanent ***
*****/

```

```

*   data in3.ZZZ;
*       set bs_reglg;
*   run;

data time;
  set time;
  format stop datetime16.;
  stop=datetime();
  output;
run;

proc print data=time;
  title 'Temps requis pour exécuter le programme';
run;

/* Fin du programme SAS Bootvar */

```