

# Travaux pratiques avec le logiciel SAS

Échantillonnage  
STT-2000

Audrey Béliveau & David Haziza

Département de mathématiques et de statistique  
Université de Montréal

Automne 2010

## TABLE DES MATIERES

Fiche 1 .....	4
1.1 Créer un tableau SAS .....	4
1.2 Importer un jeu de données .....	5
1.3 Sauvegarder un jeu de données dans ses dossiers personnels.....	6
1.4 Travailler sur les colonnes .....	7
1.5 Travailler sur les lignes .....	9
Fiche 2 .....	11
2.1 La procédure PRINT .....	11
2.2 La procédure SORT .....	12
2.3 La procédure FREQ .....	12
2.4 La procédure MEANS .....	13
2.5 La procédure GPLOT.....	15
2.6 La procédure GCHART .....	16
2.7 La procédure BOXPLOT .....	16
Fiche 3 .....	18
3.1 La procédure SURVEYSELECT .....	18
3.2 La procédure SURVEYMEANS .....	21
3.3 Exercice – Effet de la taille échantillonnale sur la variance de l’estimateur en EASSR ....	23
Fiche 4 .....	26
4.1 Exercice – L’intervalle de confiance en échantillonnage .....	26
4.2 Exercice – Distribution de la taille d’échantillon pour le plan Bernouilli .....	29
4.3 Exercice – Comparaison de l’efficacité des plans EASSR et Bernouilli .....	32
4.4 Exercice – Comparaison de l’efficacité des plans EASSR et systématique .....	35
Fiche 5 .....	38
5.1 Sélection d’un échantillon stratifié et estimation .....	38
5.2 Exercice – Comparaison des répartitions de Neyman et proportionnelle.....	41
5.3 Exercice – Comparaison de l’EASSR et du plan stratifié.....	43
Fiche 6 .....	46
6.1 Estimation du total d'un domaine.....	46
6.2 Estimation d'un ratio .....	48
6.3 Estimation de la moyenne d'un domaine.....	51
6.4 Exercice – Effet d’une petite taille d’échantillon sur le biais de l’estimateur d’un ratio....	54

Fiche 7 .....	57
7.1 La procédure REG .....	57
7.2 Estimation en présence d'information auxiliaire .....	60
7.3 Exercice – Comparaison des estimateurs par le ratio, la regression et la différence .....	64
ANNEXE A - Les opérateurs en SAS.....	69
ANNEXE B - Les études par simulation en théorie de l'échantillonnage.....	70
ANNEXE C - Macro pour la répartition .....	75

## Fiche 1

Lors des séances de travaux pratiques qui suivront, vous serez amenés à vous familiariser avec le **logiciel SAS**. Il s'agit d'un logiciel d'étude de données permettant notamment de manipuler des fichiers de données et d'utiliser un ensemble de procédures statistiques produisant des résultats.

Afin que le système SAS puisse manipuler les données, elles doivent être mises sous forme de **tableau SAS**, un fichier dont l'extension est .sas7bdat. Les tableaux SAS peuvent être temporaires ou permanents. Par défaut, les tableaux SAS sont temporaires et sont sauvegardés dans le Work.

Les programmes SAS sont constitués d'étapes DATA et PROC. Les **étapes DATA** préparent les tableaux SAS alors que les **étapes PROC** les exploitent. Cette première fiche vous permettra de vous familiariser avec l'environnement SAS et la manipulation des jeux de données à l'aide des étapes DATA.

### 1.1 Créer un tableau SAS

Une première façon d'obtenir un tableau de données en SAS est de taper les données à la main, directement dans le logiciel. En utilisant une étape DATA, on va créer un tableau SAS nommé « infoperso ». L'**instruction INPUT** contient le nom des variables dont les valeurs seront entrées. Le signe \$ spécifie que la variable qui précède est de nature alphanumérique. Ce tableau se trouvera dans le Work.

```
data infoperso;
    input id prenom $ sexe $ age;
    cards;
    1 Josee F 22
    2 Michael H 64
    3 Cedric H 37
    4 Nicole F 32
run;
```

/\*Il arrive souvent de vouloir générer des **variables** qui sont **aléatoires**. Ci-dessous, nous générons un tableau contenant dix observations de variables aléatoires uniformes(-1,1), binomiales(10,1/2) et normales(10,4).\*/

```
data va;
  do i = 1 to 10;
    uniforme = ranuni(999)*2-1;
    binomiale = ranbin(999,10,0.5);
    normale = rannor(999)*4+10;
  output;
end;
run;
```

## 1.2 Importer un jeu de données

Nous verrons comment **importer un jeu de données** dans SAS. Le jeu de données peut être donné sous plusieurs formes, mais nous traiterons seulement les formats *.txt* et *.sas7bdat* (table SAS). Lors des prochaines séances, tous les jeux de données nécessaires vous seront déjà fournis sous forme de table SAS.

### 1.2.1 Importer un jeu de données de format .txt dans le Work

/\*En utilisant une étape DATA, on va créer un tableau SAS nommé « infos » à partir du fichier infos.txt. L'**instruction INFILE** permet de spécifier l'emplacement du fichier texte et l'**option DELIMITER** spécifie le délimiteur utilisé pour séparer les données dans le fichier texte.

L'**instruction INPUT** contient les variables qui seront lues. Le signe \$ spécifie que la variable qui précède est de nature alphanumérique. Par défaut, SAS garde au maximum huit caractères lors de la lecture, mais on peut changer ce nombre en rajoutant :?. après le signe \$, où ? représente le nombre de caractères voulus.\*/

```
data infos;  
    infile "stt2000/tp1/infos.txt" delimiter=';';  
    input numero position $ prenom $ :11. nom $ :11. ;  
run;
```

### 1.2.2 Importer un tableau SAS dans le Work

*/\*L'instruction LIBNAME permet de fournir à SAS le nom d'une librairie ainsi que son emplacement. Cette librairie devient accessible dans la fenêtre Explorer.\*/*

```
libname mylib "stt2000/tp1/ ";
```

*/\*On crée dans le Work un tableau SAS nommé « stats » dans lequel on copie le contenu du tableau nommé « stats » situé dans la librairie mylib.\*/*

```
data stats; *on aurait pu écrire: data Work.stats;  
    set mylib.stats;  
run;
```

### 1.3 Sauvegarder un jeu de données dans ses dossiers personnels

Il est également possible de sauvegarder une copie d'un tableau SAS dans un répertoire de ses dossiers personnels.

*/\*On crée dans la librairie « mylib » un tableau SAS nommé « infos » dans lequel on copie le contenu du tableau nommé « infos » situé dans le Work. Ce tableau est donc sauvegardé à l'emplacement "stt2000/tp1".\*/*

```
data mylib.infos;  
    set infos;  
run;
```

## 1.4 Travailler sur les colonnes

Nous allons voir le genre de travail que l'on peut effectuer sur **les colonnes** d'un tableau. En fait, dans une table SAS, les colonnes contiennent les **variables** ; les lignes contiennent les observations. Nous verrons comment ajouter ou enlever des variables. Nous verrons également comment fusionner deux tableaux SAS et nous expliquerons ce que signifie le mot « fusionner ».

### 1.4.1 Ajouter des variables

/\*On ajoute deux colonnes au tableau SAS nommé « stats »: la première (points) contient la somme du nombre de buts et d'aides pour chaque joueur, la seconde (efficacite) contient le pourcentage de buts marqués rapporté au nombre de lancers. Vous pouvez vous référer à l'annexe A pour une explication des symboles +, /,\* et plusieurs autres.\*/

```
data stats;  
    set stats;  
    points=buts+aides;  
    efficacite=buts/lancers*100;  
run;
```

/\*On ajoute au tableau la colonne « classe », qui classe les joueurs dans des catégories selon leur nombre de points. Vous pouvez vous référer à l'annexe A pour les symboles LE, LT, GE et plusieurs autres.\*/

```
data stats;  
    set stats;  
    if 0 LE points LT 20 then classe=4;  
    else if 20 LE points LT 40 then classe=3;  
    else if 40 LE points LT 60 then classe=2;  
    else if points GE 60 then classe=1;  
run;
```

### 1.4.2 Enlever des variables

/\*On peut demander à SAS d'enlever les variables « buts », « aides », « lancers » et « efficacite » du tableau « stats »\*/

```
data stats;  
    set stats;  
    drop buts aides lancers efficacite;  
run;
```

/\*Le même résultat aurait pu être obtenu avec le code suivant, qui consiste à garder les variables « prenom », « nom », « parties », « points », « punitions » et « classe ».\*/\*

```
/*data stats2;  
    set stats;  
    keep prenom nom parties points punitions classe;  
run;*/
```

### 1.4.3 Fusionner deux tableaux SAS

/\*La **fusion** de deux tableaux SAS consiste à combiner les observations des deux tableaux en se basant sur les valeurs de certaines **variables communes**.

La première étape essentielle avant de pouvoir fusionner les tableaux est de les **trier selon les variables communes utilisées pour la fusion** en utilisant la **procédure SORT**. Par défaut, les données sont triées dans l'ordre croissant.\*/\*

```
proc sort data=stats;  
    by nom;  
run;
```

/\*Nous n'avons pas trié le tableau infos, car il l'était déjà. On parvient à la fusion des tableaux « infos » et « stats » dans un tableau nommé « Canadiens » en utilisant une **étape DATA** et l'**instruction MERGE**.\*/

```
data Canadiens;  
    merge infos stats;  
run;
```

## 1.5 Travailler sur les lignes

Nous allons voir le travail que l'on peut entreprendre sur **les lignes**. Rappelons que les lignes d'une table SAS représentent les **observations**. Nous allons voir comment supprimer des lignes. Nous verrons également comment concaténer deux tables SAS et nous expliquerons la signification du mot « concaténer ».

### 1.5.1 Supprimer des lignes

*/\*On ne garde que les joueurs de centre dans un tableau SAS qu'on appelle « centres ».\*/\**

```
data centres;  
    set Canadiens;  
    if position="C";  
run;
```

*/\*On ne garde que les ailiers dans un tableau SAS qu'on appelle ailiers.\*/\**

```
data ailiers;  
    set Canadiens;  
    if position="L" OR position="R";  
run;
```

*/\*On veut garder seulement les trois meilleurs joueurs de centre dans un tableau. D'abord, on trie les joueurs de centre par rapport à leur nombre de points. On ajoute **DESCENDING** devant la variable à trier pour demander un tri par ordre décroissant.\*/\**

```
proc sort data=centres;  
    by descending points;
```

```
run;
```

/\*On sélectionne les trois premières observations du tableau trié pour les mettre dans un nouveau tableau qu'on appelle « top3centres ».\*/\*

```
data top3centres;
```

```
    set centres(obs=3);
```

```
run;
```

/\*Le même résultat aurait pu être obtenu avec le code suivant, qui consiste à sélectionner les joueurs ayant un pointage supérieur ou égal à 39.\*/\*

```
data top3centres2;
```

```
    set centres(where=(points>=39));
```

```
run;
```

### 1.5.2 Concaténer deux tableaux

/\*La **concaténation** de deux tableaux consiste à combiner les tableaux en les **disposant l'un en dessous de l'autre** dans un seul tableau. Le nombre d'observations du nouveau tableau est donc la somme du nombre d'observations dans chacun des deux tableaux de départ.

La concaténation se fait en inscrivant, l'un après l'autre, les noms des tableaux à concaténer dans l'**instruction SET** d'une **étape DATA**.\*/

```
data attaquants;
```

```
    set centres ailiers;
```

```
run;
```

## Fiche 2

Au cours de cette démonstration, quelques procédures (étapes PROC) élémentaires de SAS vous seront présentées:

- PROC PRINT qui permet d'imprimer un tableau SAS ;
- PROC SORT, qui permet de trier le tableau par rapport à certaines colonnes ;
- PROC FREQ, qui permet de produire des tables de fréquences ou des tableaux de contingence ;
- PROC MEANS qui permet d'obtenir des statistiques descriptives.

Lorsqu'une procédure est appelée, les résultats produits par SAS sont imprimés par défaut dans la fenêtre Output. Nous verrons qu'il est également possible d'imprimer les résultats dans des fichiers *pdf* ou *html* qui pourront être sauvegardés dans vos dossiers personnels.

Des procédures graphiques seront également présentées :

- PROC GPLOT qui permet d'obtenir un graphique de nuage de points ;
- PROC GCHART qui permet d'obtenir un histogramme ;
- PROC BOXPLOT qui permet d'obtenir un boxplot ou des « side-by-side » boxplots.

Cette fiche utilisera le tableau SAS nommé Canadiens qu'il faut d'abord importer dans le Work.

```
libname mylib "st2000/tp2/ ";
```

```
data Canadiens;
```

```
    set mylib.Canadiens ;
```

```
run;
```

### 2.1 La procédure PRINT

La forme de base de la **procédure PRINT** permet **d'imprimer un tableau SAS** dans la **fenêtre Output**. Il est possible de lui ajouter diverses options et instructions.

/\*Voici la **forme de base** de la procédure PRINT\*/

```
proc print data=Canadiens;  
run;
```

/\*L'**option OBS** permet de spécifier le nombre d'observations à imprimer et l'**instruction VAR** permet de spécifier les variables à imprimer\*/

```
proc print data=Canadiens (obs=15);  
    var prenom nom points;  
run;
```

## 2.2 La procédure SORT

La **procédure SORT** permet de **trier** un tableau SAS par rapport à certaines colonnes.

/\*L'**instruction BY** permet de spécifier les variables par rapport auxquelles ordonner. Par défaut, l'ordre croissant est utilisé, mais il suffit de rajouter le mot **descending** devant la variable en question pour que l'ordre soit décroissant.\*/

```
proc sort data=Canadiens;  
    by nom prenom;  
run;
```

## 2.3 La procédure FREQ

La forme de base de la **procédure FREQ** permet de produire un **tableau de fréquences** pour chaque variable d'un tableau SAS. Il est possible de lui ajouter diverses options et instructions afin de produire, par exemple, des **tableaux de contingence**.

/\*À partir de maintenant, imprimons les sorties dans un fichier *pdf*.\*/

```
ods pdf file="stt2000/tp2/outpdf";
```

*/\*Voici la procédure FREQ de base\*/*

```
proc freq data=Canadiens;  
run;
```

*/\*L'instruction TABLES permet de spécifier pour quelles variables les tableaux de fréquences ou de contingence sont requis\*/*

```
proc freq data=Canadiens;  
    tables position position*classe;  
run;
```

*/\*L'option NOPRINT spécifie de ne pas imprimer les résultats (ni dans la fenêtre Output, ni dans le fichier pdf) et l'option OUT= permet d'écrire les résultats du dernier tableau de fréquences ou de contingence dans un jeu de données SAS\*/*

```
proc freq data=Canadiens noprint;  
    tables position position*classe/ out=outfreq;  
run;
```

*/\*N'oublions pas de fermer le fichier pdf; il sera ainsi sauvegardé dans le dossier demandé.\*/*

```
ods pdf close;
```

## 2.4 La procédure MEANS

La **procédure MEANS** permet de produire des **statistiques descriptives** comme le nombre d'observations, la moyenne, l'écart-type. Celles-ci, de même que plusieurs autres, vous seront présentées. Il est également possible d'ajouter diverses options et instructions à la procédure.

*/\*À partir de maintenant, imprimons les sorties dans un fichier html\*/*

```
ods html file="stt2000/tp2/outhtml";
```

/\*La procédure **MEANS de base** produit les statistiques descriptives suivantes pour toutes les variables: nombre d'observations (N), moyenne (MEAN), écart-type (STD), minimum (MIN) et maximum (MAX).\*/

```
proc means data=Canadiens;  
run;
```

/\*On peut spécifier en option les statistiques descriptives voulues. Quelques statistiques pertinentes pourraient être: la somme (SUM), la variance (VAR), l'étendue (RANGE) ou le coefficient de variation (CV).

On peut aussi utiliser l'**instruction VAR** pour spécifier les variables pour lesquelles les statistiques descriptives sont requises.\*/

```
proc means data=Canadiens n mean sum var range cv;  
    var parties points punitions;  
run;
```

/\*On ajoute l'**instruction CLASS** pour demander les statistiques descriptives pour chaque valeur prise par la variable suivant le mot CLASS.\*/

```
proc means data=Canadiens n mean std range;  
    var parties points punitions;  
    class position;  
run;
```

/\*L'**option NOPRINT** spécifie de ne pas imprimer les résultats (ni dans la fenêtre Output, ni dans le fichier *pdf*). Cette option ne peut être spécifiée que si l'**instruction OUT=** est utilisée. Cette dernière permet d'écrire les statistiques descriptives demandées dans un tableau SAS.

Dans le code suivant, ce tableau est appelé « outmeans » et les statistiques demandées sont le nombre maximal de parties jouées (maxParties), le nombre total de points (totPoints) et le nombre total de punitions (totPunitions).\*/

```
proc means data=Canadiens noprint;
    var parties points punitions;
    class position;
    output out=outmeans max=maxParties sum(points punitions)=totPoints totPunitions;
run;

/*N'oublions pas de fermer le fichier html.*/

ods html close;
```

## 2.5 La procédure GPLOT

La **procédure GPLOT** permet de produire un graphique de nuage de points. Il est possible d'ajouter des options et instructions à la procédure afin de choisir l'emplacement de sauvegarde du graphique dans les dossiers personnels, nommer les axes, choisir la forme et la couleur des points et ajouter un titre au graphique.

```
goptions reset=all device=gif gsfname=output gsfmode=replace;
filename output 'stt2000/tp2/graphique1.gif';
axis1 label=(a=90 'Nombre de points');
axis2 label=('Nombre de parties');
symbol1 value=circle color=red;

proc gplot data=Canadiens;
    plot1 points*parties/ vaxis=axis1 haxis=axis2;
    title "Nombre de points selon le nombre de parties jouees";
run;
```

## 2.6 La procédure GCHART

La **procédure GCHART** permet de produire un histogramme. Il est possible d'ajouter des options et instructions à la procédure afin de choisir l'emplacement de sauvegarde du graphique dans les dossiers personnels, nommer les axes et ajouter un titre au graphique.

```
goptions reset=all device=gif gsfname=output gsfmode=replace;
filename output 'stt2000/tp2/histogramme1.gif';
axis1 label=(angle=90 'Frequence');
axis2 label=('Nombre de points');

proc gchart data=Canadiens;
vbar points/ raxis=axis1 maxis=axis2;
title "Distribution du nombre de points";
run;
```

## 2.7 La procédure BOXPLOT

La **procédure BOXPLOT** permet de produire un boxplot ou des « side-by-side » boxplots. Il est possible d'ajouter des options et instructions à la procédure afin de choisir l'emplacement de sauvegarde du graphique dans les dossiers personnels, nommer les axes et ajouter un titre au graphique. Par défaut, la procédure utilise toutes les observations pour produire le boxplot sans faire de distinction pour les valeurs aberrantes. L'option **BOXSTYLE=schematic** permet d'exclure ces valeurs aberrantes du boxplot et de les afficher avec un symbole choisi avec l'option **IDSYMBOL** (par exemple, **IDSYMBOL=circle** pour le cercle).

```
proc sort data=Canadiens;
by position;
run;

goptions reset=all device=gif gsfname=output gsfmode=replace;
filename output 'stt2000/tp2/boxplots1.gif';
axis1 label=('Nombre de points');
```

```
axis2 label=('Position');
```

```
proc boxplot data=Canadiens;
```

```
plot points*position / vaxis=axis1 haxis=axis2 boxstyle=schematic idsymbol=circle;
```

```
title "Distribution du nombre de points selon la position";
```

```
run;
```

### Fiche 3

Lors de cette séance de travaux pratiques, nous présenterons d'abord deux procédures utiles dans le domaine de l'échantillonnage: SURVEYSELECT, pour la sélection d'échantillons et SURVEYMEANS pour l'estimation de paramètres. Ensuite, nous ferons l'exemple 2.4 des notes de cours, qui permettra d'illustrer l'effet de la taille échantillonnale sur la variance des estimateurs pour l'EASSR.

#### 3.1 La procédure SURVEYSELECT

La procédure SURVEYSELECT offre à l'utilisateur un choix de méthodes probabilistes pour **sélectionner un échantillon** aléatoire à partir d'une base de sondage organisée sous forme d'une table SAS. La procédure peut sélectionner, par exemple, un échantillon aléatoire simple sans remise ou un échantillon systématique (mais pas Bernouilli). L'échantillon obtenu peut être conservé sous forme d'une table SAS. On peut également, avec SURVEYSELECT, sélectionner des **échantillons répliqués**, c'est-à-dire un nombre  $R$  d'échantillons tirés dans la même population selon le même plan de sondage. Ces échantillons sont placés les uns en dessous des autres dans une table SAS.

Voici quelques options de la procédure SURVEYSELECT :

<b>Option</b>	<b>Fonction</b>
DATA=	Désigner la table de données en entrée contenant la base de sondage
METHOD=	Spécifier la méthode d'échantillonnage (ex : SRS, SYS)
SAMPsize= ou N=	Spécifier la taille de l'échantillon
OUT=	Désigner la table en sortie contenant l'échantillon
STATS	Conserver dans la table en sortie les probabilités d'inclusion simple et les poids de sondage par unité quand le sondage est auto-pondéré.
OUTSIZE	Conserver dans la table en sortie la taille de la population et celle de l'échantillon.
NOPRINT	Supprimer l'impression des résultats.
REP=	Spécifier le nombre d'échantillons répliqués.

### 3.1.1 Tirage d'un échantillon aléatoire simple sans remise

Nous verrons comment tirer un **EASSR** avec la procédure SURVEYSELECT. Il faudra spécifier l'option **METHOD=SRS**.

```
/*On importe le jeu de données*/
```

```
libname mylib "stt2000/tp3/ ";
```

```
data pop_paroisses;
```

```
    set mylib.pop_paroisses(keep=paroisse naissance deces);
```

```
run;
```

```
/*On tire l'échantillon*/
```

```
title "Sondage aleatoire simple sans remise";
```

```
proc surveyselect data=pop_paroisses method=srs n=30 stats out=echsrs ;
```

```
run;
```

Sondage aleatoire simple sans remise	
The SURVEYSELECT Procedure	
Selection Method	Simple Random Sampling
Input Data Set	POP_PAROISSES
Random Number Seed	91954857
Sample Size	30
Selection Probability	0.142857
Sampling Weight	7
Output Data Set	ECHSRS

```
/*On imprime les premières observations dans la fenêtre Output*/
```

```
title2 "Extrait de l'échantillon";
```

```
proc print data=echsrs (obs=10);
```

```
run;
```

Sondage aleatoire simple sans remise Extrait de l'échantillon					
Obs	paroisse	naissance	deces	Selection Prob	Sampling Weight
1	Notre-Dame-de-Portneuf	17	12	0.14286	7
2	Notre-Dame-des-Anges	0	1	0.14286	7
3	Packington	6	4	0.14286	7
4	Saint-Alexis-de-Matapédia	5	3	0.14286	7
5	Saint-Barthélémy	9	15	0.14286	7
6	Saint-Clément	7	7	0.14286	7
7	Saint-Cléophas	5	4	0.14286	7
8	Sainte-Angèle-de-Monnoir	24	4	0.14286	7
9	Sainte-Anne-de-la-Pocatière	1	0	0.14286	7
10	Sainte-Anne-de-Sabrevois	23	12	0.14286	7

### 3.1.2 Tirage d'un échantillon systématique

Nous verrons comment tirer un **échantillon systématique** avec la procédure SURVEYSELECT. Il faudra spécifier l'option **METHOD=SYS**.

/\*On tire l'échantillon\*/

title "Tirage systématique";

proc surveyselect data=pop\_paroisses method=sys n=30 out=echsys outsize;

run;

Tirage systématique	
The SURVEYSELECT Procedure	
Selection Method	Systematic Random Sampling
Input Data Set	POP_PAROISSES
Random Number Seed	310859812
Sample Size	30
Selection Probability	0.142857
Sampling Weight	7
Output Data Set	ECHSYS

/\*On imprime les premières observations dans la fenêtre Output\*/

```
title2 "Extrait de l'échantillon";  
proc print data=echsyst (obs=10);  
run;
```

Tirage systématique Extrait de l'échantillon					
Obs	paroisse	naissance	deces	Total	Sample Size
1	Grande-Vallée	14	6	210	30
2	Matapédia	7	6	210	30
3	Notre-Dame-de-Stanbridge	6	10	210	30
4	Saint-Adelme	1	8	210	30
5	Saint-Anaclet-de-Lessard	36	11	210	30
6	Saint-Augustin-de-Woburn	9	0	210	30
7	Saint-Christophe-d'Arthabaska	30	4	210	30
8	Saint-Damase	4	4	210	30
9	Saint-Donat	13	13	210	30
10	Sainte-Apolline-de-Patton	6	7	210	30

### 3.2 La procédure SURVEYMEANS

La procédure SURVEYMEANS fournit, à partir d'un échantillon donné, des **estimations telles que la moyenne ou le total d'une variable d'intérêt** dans la population. Elle peut aussi fournir **l'estimateur de la variance** ou l'estimation d'un intervalle de confiance (utilisant la distribution  $t$  toutefois). Contrairement à la procédure SURVEYSELECT de sélection de l'échantillon, on ne déclare pas dans la procédure SURVEYMEANS la méthode de tirage utilisée. Pour effectuer ses estimations, la procédure SURVEYMEANS fait plutôt appel aux **ponds de sondage** qui dépendent naturellement de la technique de sélection utilisée.

Pour estimer un total dans un plan de sondage tel l'EASSR, le BE ou le SY, la procédure SURVEYMEANS utilise l'estimateur de Horvitz-Thompson :

$$\hat{t}_y = \sum_{i \in s} \frac{1}{\pi_i} y_i = \sum_{i \in s} w_i y_i.$$

Pour estimer une moyenne, la procédure SURVEYMEANS utilise

$$\tilde{y}_s = \frac{\hat{t}_y}{\hat{N}},$$

où  $\hat{N} = \sum_{i \in s} w_i$ . Lorsque le plan de sondage est auto-pondéré et à taille fixe, on a  $\hat{N} = N$

(preuve laissée en exercice) et donc  $\tilde{y}_s = \bar{y}_s$ .

Afin que l'estimateur de la variance soit correctement calculé, il ne faut pas oublier d'utiliser **l'option N=** qui spécifie la taille de la population,  $N$ . Le logiciel estime la variance d'un total dans un sondage systématique comme celle d'un sondage aléatoire simple sans remise avec :

$$\hat{V}(\hat{t}_y) = N^2 \left(1 - \frac{n}{N}\right) \frac{s_y^2}{n}.$$

Si la taille de la population n'est pas spécifiée en option, le facteur de correction  $\left(1 - \frac{n}{N}\right)$  pour population finie ne sera pas apporté.

Voici un exemple de l'utilisation de PROC SURVEYMEANS dans le cas d'un EASSR. **L'option ODS OUTPUT STATISTICS=** permet de créer une table SAS contenant les estimations.

```
title "Tirage aléatoire simple sans remise";
title2 "Estimation de paramètres";
proc surveymeans data=echsrs N=210 mean var clm /*sum varsum clsum*/;
    var naissance;
    weight SamplingWeight;
    ods output Statistics=estimations;
run;
title; /*pour enlever les titres*/
```

Tirage aléatoire simple sans remise				
Estimation de paramètres				
The SURVEYMEANS Procedure				
Data Summary				
Number of Observations				30
Sum of Weights				210
Statistics				
Variable	Mean	Std Error of Mean	Var of Mean	95% CL for Mean
naissance	9.633333	1.397029	1.951691	6.77608728 12.4905794

### 3.3 Exercice – Effet de la taille échantillonnale sur la variance de l’estimateur en EASSR

Nous allons reproduire l’exemple 2.4 des notes de cours (p. 29). Nous allons aller un peu plus loin en calculant l’espérance et la variance Monte Carlo de l’estimateur.

```
/*On importe le jeu de données*/
```

```
data pop;
    set mylib.pop;
run;
```

```
/*On vérifie que la moyenne des y dans la population est 100*/
```

```
proc means data=pop mean;
    var y;
run;
```

```
/*On sélectionne les R=100 échantillons répliqués par EASSR*/
```

```
proc surveyselect data=pop method=srs n=50 rep=100 stats out=echrep;  
run;
```

```
/*On estime la moyenne des y*/
```

```
proc means data=echrep noprint;  
    var y;  
    by replicate;  
    weight SamplingWeight;  
    output out=estimations mean=moy_y_ech;  
run;
```

```
/*On calcule l'erreur relative*/
```

```
data estimations;  
    set estimations;  
    err_rel=(moy_y_ech-100)/100*100;  
run;
```

```
/*On produit le graphique*/
```

```
goptions reset=all device=gif gsfname=output gsfmode=replace;  
filename output 'stt2000/tp3/graphique1.gif';  
axis1 order = (-15 to 15 by 3) label=(a=90 'Erreur relative (en %)');  
axis2 label=('Numero de l echantillon');  
symbol1 value=circle color=red interpol=join;  
proc gplot data=estimations;  
    plot1 err_rel*replicate/ vref=0 vaxis=axis1 haxis=axis2;  
    title "Erreur relative avec n=50";  
run;
```

```
/*On calcule l'espérance et la variance Monte Carlo*/
```

```
proc means data=estimations mean noprint;
```

```

var moy_y_ech;
output out=espMC mean=espMC;
run;

data _NULL_; *permet d'être dans une étape DATA sans créer de table SAS;
set espMC;
call symput('espMC',espMC); /*permet de créer une variable globale contenant la
                                valeur de espMC*/

run;

data estimations;
set estimations;
err_rel2=(moy_y_ech-&espMC)**2; /*il faut ajouter le signe & devant la variable
                                globale*/

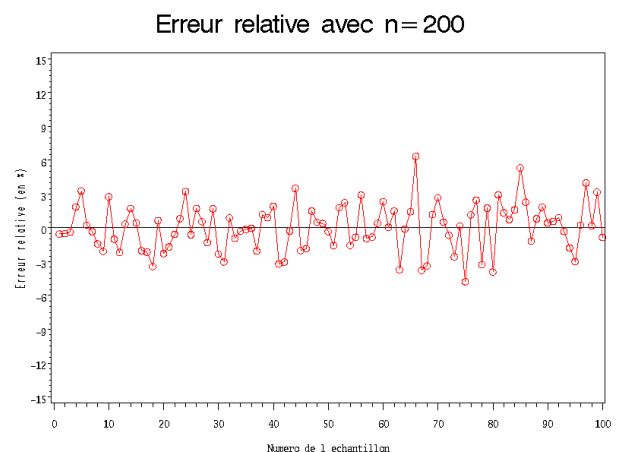
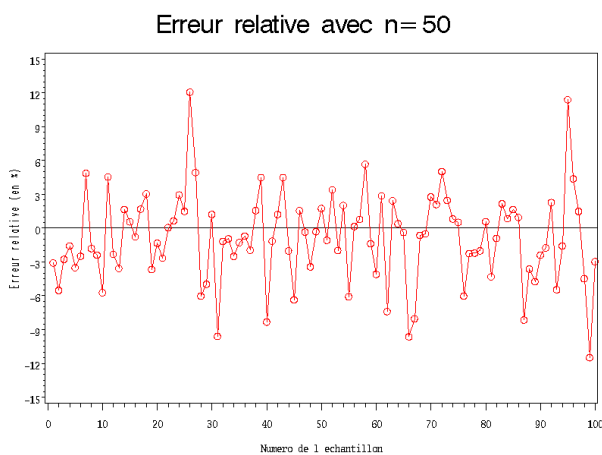
run;

proc means data=estimations mean;
var err_rel2;

run;

```

Tableau des résultats		
$n$	$E_{MC}$	$V_{MC}$
50		
200		



## Fiche 4

Cette fiche vous propose quatre exercices portant sur les plans de sondage EASSR, BE et SY. Le premier exercice servira à illustrer le concept d'intervalle de confiance en échantillonnage; le second, à examiner la distribution de la taille aléatoire pour le plan BE; le troisième, à comparer les plans EASSR et BE et finalement le quatrième, à comparer l'efficacité des estimateurs pour les plans EASSR et SY.

### 4.1 Exercice – L'intervalle de confiance en échantillonnage

Cet exercice porte sur l'**interprétation de l'intervalle de confiance** en échantillonnage. En effet, il est important de savoir différencier le concept d'intervalle de confiance dans le cadre de populations finies ou infinies (voir section 2.4.2 des notes de cours).

a)

Pour illustrer le concept d'intervalle de confiance en échantillonnage, nous considérerons **tous les échantillons possibles** de taille  $n = 4$  parmi une population (pop1) de taille  $N = 15$  par l'EASSR. Dans chaque échantillon, on calculera l'estimateur du total,  $\hat{t}_y$ , et l'estimateur de la variance

$$\hat{V}(\hat{t}_y) = N^2 \left(1 - \frac{n}{N}\right) \frac{s_y^2}{n}.$$

Nous construirons un intervalle de confiance de niveau  $\alpha = 5\%$  dans chaque échantillon donné par :

$$\hat{t}_y \pm 1.96 \sqrt{\hat{V}(\hat{t}_y)}.$$

Pour chaque intervalle, nous déterminerons si celui-ci contient la vraie valeur du total  $t_y = 150$ . Ainsi, nous pourrons déterminer le niveau de confiance **exact** de l'IC en calculant la proportion des échantillons qui contiennent la vraie valeur du paramètre  $t_y$  parmi tous les échantillons possibles.

`libname mylib "stt2000/tp4/";`

```
data pop_allsamples;  
    set mylib.pop_allsamples;  
run;
```

/\*On peut vérifier que le total de y dans la population est 150\*/

```
proc means data=pop_allsamples sum;  
    var y;  
run;
```

/\*On calcule l'estimateur du total et l'écart-type estimé pour chaque échantillon\*/

```
data allsamples;  
    set mylib.allsamples;  
run;
```

ods select none; /\*afin de ne plus imprimer les sorties dans la fenêtre output\*/

```
proc surveymeans data=allsamples N=15 sum;  
    var y;  
    by replicate;  
    weight SamplingWeight;  
    ods output Statistics=estimations;
```

```
run;
```

ods select all; /\*afin d'imprimer de nouveau les sorties dans la fenêtre output\*/

/\*On calcule l'IC et on définit une variable indicatrice de présence du vrai total dans l'IC\*/

```
data estimations;  
    set estimations;  
    LowerCLSum=sum-1.96*StdDev;  
    UpperCLSum=sum+1.96*StdDev;  
    if LowerCLSum<150 AND UpperCLSum>150 then Id_IC=1;  
    else Id_IC=0;  
run;
```

/\*On calcule le vrai taux de couverture\*/

```
proc means data=estimations mean;  
    var Id_IC;  
run;
```

Tableau des résultats	
Vrai taux de couverture	Taux de couverture théorique
%	95 %

b)

Comme on dispose de tous les échantillons possibles, on peut aussi s'intéresser au **vrai biais** et à la **vraie variance** de l'estimateur du total. Le vrai biais peut être calculé par

$$\text{Biais}(\hat{t}_y) = \sum_{s \in \Omega} \hat{t}_y p(s) - t_y = \sum_{s \in \Omega} (\hat{t}_y - t_y) p(s).$$

Cette expression se simplifie dans le cas de l'EASSR pour donner zéro. La vraie variance peut être calculée par

$$V(\hat{t}_y) = \sum_{s \in \Omega} (\hat{t}_y - t_y)^2 p(s),$$

expression qui est équivalente, dans le cas de l'EASSR, à

$$V(\hat{t}_y) = N^2 \left(1 - \frac{n}{N}\right) \frac{S_y^2}{n}.$$

Nous pourrions vérifier ces équivalences.

/\*On calcule le vrai biais et la vraie variance à partir de la somme sur tous les échantillons possibles.\*/

```
data estimations;  
    set estimations;  
    err=sum-150;
```

```

err_ps=err*0.0007326; *on a multiplié par p(s);
err2=err**2; *justifiable si le biais est nul;
err2_ps=err2*0.0007326; *on a multiplié par p(s);
run;

proc means data=estimations sum;
    var err_ps err2_ps;
    output out=Vraies_proprietes(drop=_TYPE_ _FREQ_) mean=Vrai_biais
    Vraie_variance;
run;

```

/\*On calcule  $S_y^2$  dans la population. Cette valeur nous permettra de calculer facilement la vraie variance à partir de la formule pour l'EASSR.\*/

```

proc means data=pop_allsamples var;
    var y;
run;

```

Tableau des résultats		
	Vrai biais	Vraie variance
Formule $\sum_{s \in \Omega}$		
Formule EASSR	0	

#### 4.2 Exercice – Distribution de la taille d'échantillon pour le plan Bernouilli

Cet exercice concerne la **distribution de la taille d'échantillon aléatoire** pour le **plan de sondage BE**. Nous tirerons  $R = 1000$  échantillons BE avec  $\pi = 0.05$  dans une population de taille  $N = 500$ . Nous produirons par la suite un histogramme de la distribution des tailles échantillonnales  $n_s$ . Nous pourrons aussi calculer l'espérance et la variance Monte Carlo de la

taille échantillonnale et comparer les valeurs obtenues avec les valeurs théoriques pour une binomiale.

```
data pop_taille;  
    set mylib.pop_taille;  
run;
```

/\*L'échantillonnage BE ne peut pas être effectué directement avec la procédure SURVEYSELECT. Toutefois, sa programmation est simple. On commence par produire  $R = 1000$  répétitions de la population.\*/

```
proc surveyselect data= pop_taille method=srs n=500 rep=1000 out=popreps;  
run;
```

/\*Pour chaque répétition, on génère des variables aléatoires ( $e$ ) uniformes de paramètre 0 et 1 et on sélectionne dans l'échantillon les unités pour lesquelles  $e < \pi$ \*/

```
data popreps;  
    set popreps;  
    e=ranuni(1);  
    if e<0.05 then ind=1;  
    else ind=0;  
run;
```

/\*On calcule les tailles échantillonnales  $n_s$ \*/

```
proc means data=popreps noprint;  
    var ind;  
    by replicate;  
    output out=tailles sum=tailles;  
run;
```

/\*On produit l'histogramme de la distribution des tailles échantillonnales\*/

```

goptions reset=all device=gif gsfname=output gsfmode=replace;
filename output 'stt2000/tp4/histogramme.gif';
axis1 label=(angle=90 'Frequence');
axis2 label=('Taille echantillonnale');
proc gchart data=tailles;
    vbar tailles/ raxis=axis1 maxis=axis2;
    title "Distribution de la taille echantillonnale";
run;

```

/\*On calcule l'espérance et la variance Monte Carlo de la taille échantillonnale\*/

```

proc means data=tailles mean;
    var tailles;
    output out=espMC mean=espMC;
run;

```

```

data _NULL_;
    set espMC;
    call symput('espMC',espMC);
run;

```

```

data tailles;
    set tailles;
    err2=(tailles-&espMC)**2;
run;

```

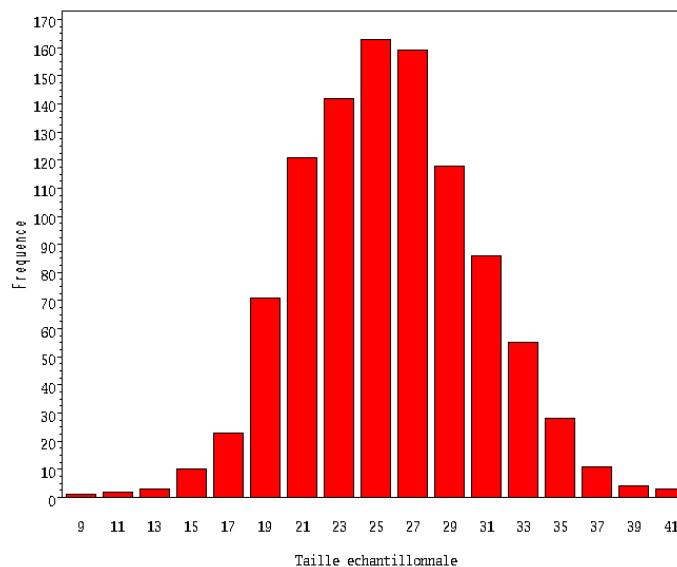
```

proc means data=tailles mean;
    var err2;
run;

```

Tableau des résultats		
	Espérance	Variance
Monte Carlo		
Théorique	$N\pi = 25$	$N\pi(1-\pi) = 23.75$

Distribution de la taille échantillonnale



### 4.3 Exercice – Comparaison de l’efficacité des plans EASSR et Bernoulli

Nous allons **comparer**, à l’aide d’une **étude par simulation**, l’efficacité des **plans de sondage aléatoire simple sans remise** et **Bernoulli** pour trois populations différentes de taille  $N = 500$  : pop1 ( $\bar{y}_U = 100$ ), pop2 ( $\bar{y}_U = 100$ ) et pop3 ( $\bar{y}_U = 0$ ). Vous avez vu que

$$deff = \frac{V_{BE}(\hat{t}_y)}{V_{EASSR}(\hat{t}_y)} = \frac{N-1}{N} + \frac{1}{CV(y)^2}.$$

Nous illustrerons donc l’effet du  $CV(y)$  sur le  $deff$ .

```
data pop1;
  set mylib.pop1;
run;
```

/\*On produit les statistiques sur la moyenne et le CV de la variable d'intérêt dans la population\*/

```
proc means data=pop1 mean std cv; /*attention: le cv donné sera en %*/  
    var y;  
run;
```

/\*On calcule la variance Monte Carlo pour un EASSR\*/

```
proc surveyselect data=pop1 method=srs n=50 rep=1000 stats out=ech_srs_reps;  
run;
```

```
proc means data=ech_srs_reps noprint;  
    var y;  
    by replicate;  
    weight SamplingWeight;  
    output out=estimations_SRS sum=tot_y_estime;  
run;
```

```
data estimations_SRS;  
    set estimations_SRS;  
    err=tot_y_estime-100*500;  
    err2=err**2; /*justifiable, car l'estimateur HT est sans biais*/  
run;
```

```
proc means data=estimations_SRS mean;  
    var err err2;  
run;
```

/\*On calcule la variance Monte Carlo pour un plan Bernouilli\*/

```
proc surveyselect data=pop1 method=srs n=500 rep=1000 out=pop_reps noprint;  
run;
```

```

data popreps;
    set pop_reps;
    e=ranuni(0);
    if e<0.1 then ind=1;
    else ind=0;
run;

data ech_BE_reps;
    set popreps(where=(ind=1));
    SamplingWeight=10;
run;

proc means data= ech_BE_reps noprint;
    var y;
    by replicate;
    weight SamplingWeight;
    output out=estimations_BE sum=tot_y_estime;
run;

data estimations_BE;
    set estimations_BE;
    err=tot_y_estime-100*500;
    err2=err**2; *justifiable, car sans biais;
run;

proc means data=estimations_BE mean;
    var err err2;
run;

```

Tableau des résultats				
	$V_{MC}$ EASSR	$V_{MC}$ BE	$deff$	$CV(y)$
pop1				9,94 %
pop2				95,79 %
pop3				- ∞

#### 4.4 Exercice – Comparaison de l'efficacité des plans EASSR et systématique

Nous allons **comparer** l'efficacité des **plans de sondage aléatoire simple sans remise** et **systématique** à l'aide d'une **étude par simulation**. La variable d'intérêt sera le nombre de naissances. Vous avez vu que

$$deff = \frac{V_{SY}(\hat{t}_y)}{V_{EASSR}(\hat{t}_y)} = \left(\frac{N-1}{N}\right) \left(1 - \frac{n}{N}\right)^{-1} [1 + (n-1)ICC].$$

Nous illustrerons donc l'effet du **degré d'homogénéité ICC** des groupes sur le  $deff$ . À partir du jeu de données des paroisses, nous pourrions ordonner les paroisses par leur nom (« paroisse ») ou par l'effectif de la population en 1999 (« pop99 ») avant de tirer l'échantillon systématique. Nous pouvons aussi ordonner la base de sondage par rapport à la variable d'intérêt (naissance), ce qui ne pourrait pas être effectué en pratique.

```

data pop_paroisses;
    set mylib.pop_paroisses(keep=paroisse naissance pop99);
run;

/*On calcule le total de y dans la population */

proc means data= pop_paroisses sum;
    var naissance;
run;

/*On calcule la variance Monte Carlo pour un EASSR*/

```

```
proc surveyselect data=pop_paroisses method=srs n=30 rep=1000 stats out=echreps;  
run;
```

```
proc means data=echreps noprint sum;  
    var naissance;  
    by replicate;  
    weight SamplingWeight;  
    output out=estimations sum=total_estime;  
run;
```

```
data estimations;  
    set estimations;  
    err=total_estime-2624;  
    err2=err**2;  
run;
```

```
proc means data=estimations mean;  
    var err2;  
run;
```

/\*On calcule la variance Monte Carlo pour un SY\*/

```
proc surveyselect data=pop_paroisses method=sys n=30 rep=1000 stats out=echreps;  
    control paroisse; *la base de sondage sera triée par rapport à cette variable;  
run;
```

```
proc means data=echreps noprint sum;  
    var naissance;  
    by replicate;  
    weight SamplingWeight;  
    output out=estimations sum=total_estime;  
run;
```

```
data estimations;
```

```

set estimations;
err=total_estime-2624;
err2=err**2;

run;

proc means data=estimations mean;
var err2;

run;

```

<b>Tableau des résultats</b>			
	$V_{MC}$ EASSR	$V_{MC}$ SY	<i>deff</i>
control: paroisse			
control: pop99			
control: naissance			

## Fiche 5

Cette fiche concerne l'échantillonnage stratifié. D'abord, les procédures SAS à utiliser pour l'échantillonnage stratifié vous seront présentées, puis deux exercices vous permettront de comparer, d'une part, les méthodes de répartition; d'autre part, l'échantillonnage stratifié avec l'EASSR.

### 5.1 Sélection d'un échantillon stratifié et estimation

En SAS, l'échantillonnage stratifié peut être réalisé avec les procédures SURVEYSELECT et SURVEYMEANS que vous connaissez.

a)

Afin de sélectionner un échantillon stratifié, il faut d'abord qu'une variable indicatrice de strates soit présente sur la base de sondage. On doit aussi **trier la base de sondage en ordre croissant** de cette variable. Par la suite, il suffit d'utiliser la procédure SURVEYSELECT avec l'**option STRATA** qui permet de désigner la variable indicatrice de strate.

```
libname mylib 'stt2000/tp5';
```

```
data pop_paroisses;  
    set mylib.pop_paroisses;  
run;
```

```
proc sort data=pop_paroisses;  
    by strate;  
run;
```

```
title "Echantillonnage stratifié";  
proc surveyselect data=pop_paroisses method=srs n=(5, 8, 5, 7) out=ech_strata stats;  
    strata strate;  
run;
```

```

Echantillonnage stratifié

The SURVEYSELECT Procedure

Selection Method      Simple Random Sampling
Strata Variable      strate

Input Data Set       POP_PAROISSES
Random Number Seed   928375346
Number of Strata     4
Total Sample Size    25
Output Data Set      ECH_STRATA

```

```

title "Extrait de l'échantillon";
proc print data=ech_strata(obs=10);
    var paroisse naissance strate;
run;

```

```

Extrait de l'échantillon

Obs    paroisse                naissance    strate
  1    Notre-Dame-Auxiliatrice-de-Buckland      8          1
  2    Packington                      6          1
  3    Saint-Alexis-de-Matapédia              5          1
  4    Saint-François-d'Assise                10         1
  5    Saint-Moise                          7          1
  6    Notre-Dame-des-Pins                   10         2
  7    Sainte-Anne-du-Sault                  3          2
  8    Saint-Édouard-de-Lotbinière           10         2
  9    Sainte-Hénédine                       11         2
 10    Saint-Jacques-le-Mineur               16         2

```

b)

Des estimations à partir de l'échantillon peuvent être obtenues avec la procédure SURVEYMEANS. Il suffit de spécifier encore une fois l'option STRATA. Ainsi, un

estimateur d'un total  $t_y = \sum_{i \in U} y_i$  est donné par

$$\hat{t}_{st} = \sum_{h=1}^H \hat{t}_h$$

et un estimateur de  $V(\hat{t}_{st})$  est donné par

$$\hat{V}(\hat{t}_{st}) = \sum_{h=1}^H \hat{V}(\hat{t}_h).$$

```

title "Estimations";
proc surveymeans data=ech_strata N=210;
    var naissance;
    strata strate / list;
    weight SamplingWeight;
    ods output Statistics=estimations;
run;
title; /*pour enlever le titre*/

```

Estimations							
The SURVEYMEANS Procedure							
Data Summary							
Number of Strata		4					
Number of Observations		25					
Sum of Weights		210					
Stratum Information							
Stratum Index	strate	Population Total	Sampling Rate	N Obs	Variable	N	
1	1	210	2.38%	5	naissance	5	
2	2	210	3.81%	8	naissance	8	
3	3	210	2.38%	5	naissance	5	
4	4	210	3.33%	7	naissance	7	
Statistics							
Variable	N	Mean	Std Error of Mean	95% CL for Mean			
naissance	25	11.896071	0.977323	9.86361744	13.9285254		

## 5.2 Exercice – Comparaison des répartitions de Neyman et proportionnelle

Cet exercice permettra de **comparer la répartition de Neyman et la répartition proportionnelle** en termes de variance. Pour ce faire, nous effectuerons une étude Monte Carlo pour estimer le total des naissances dans la population des paroisses avec un échantillon de taille  $n = 25$ . Pour une méthode de répartition donnée, nous calculerons d'abord les tailles échantillonnales dans chaque strate  $n_h$ . Le code SAS pour cette étape se révèle assez compliqué, car il faut tenir compte de certaines **contraintes pratiques** qui n'ont pas été discutées dans les notes de cours. D'une part, il est possible en utilisant la formule pour la répartition de Neyman (ou celle pour la répartition optimale) d'obtenir  $n_h > N_h$  pour un ou plusieurs  $h$ , ce qui constitue un réel problème. Supposons, par exemple, qu'on ait obtenu  $n_1 > N_1$ . Dans ce cas, on prend  $n_1 = N_1$  et on alloue de nouveau  $n_2, \dots, n_H$  en utilisant:

$$n_h = (n - N_1) \frac{N_h S_{yh}}{\sum_{h=2}^H N_h S_{yh}}.$$

Encore une fois, il est possible d'obtenir  $n_h > N_h$  pour un ou plusieurs  $h$ . Supposons, par exemple, qu'on ait obtenu  $n_2 > N_2$ . On prend alors  $n_2 = N_2$  et on alloue de nouveau  $n_3, \dots, n_H$  en utilisant:

$$n_h = (n - N_1 - N_2) \frac{N_h S_{yh}}{\sum_{h=3}^H N_h S_{yh}}.$$

On continue ce processus itératif jusqu'à ce que  $n_h \leq N_h$  pour tout  $h$ . D'autre part, il faut arrondir les tailles  $n_h$  correctement de sorte que leur somme donne  $n$ . Pour ce faire, on arrondit d'abord toutes les tailles  $n_h$  à l'entier le plus près. On notera ces nouvelles tailles  $(n_h)$ . Puis, on calcule

$$diff = n - \sum_{h=1}^H (n_h).$$

On veut  $diff = 0$ . Mais si on obtient plutôt  $diff > 0$ , il faut ajouter une unité aux plus petites tailles  $n_h \neq N_h$  de façon à obtenir  $diff = 0$ . Autrement, si on a  $diff < 0$ , il faut enlever une unité aux plus grandes tailles  $n_h \neq N_h$  de façon à obtenir  $diff = 0$ .

Une fois la répartition de la taille échantillonnale complétée, nous pourrions tirer  $R = 1000$  échantillons répliqués et estimer la variance Monte Carlo du total des naissances par l'*EQM* Monte Carlo.

```
/*On compile le code de la macro pour la répartition situé en annexe C, puis on lance la macro*/
```

```
%repartition(pop=pop_paroisses, type=neyman, idstrate=strate, interet=naissance, n=25);
```

```
/*On tire les échantillons répliqués*/
```

```
proc surveyselect data=pop_paroisses method=srs sampsize=work.allocation2 rep=1000  
out=ech_strata noprint;  
    strata strate;  
run;
```

```
/*On calcule le biais relatif et la variance Monte Carlo*/
```

```
proc sort data=ech_strata;  
    by replicate strate;  
run;
```

```
ods select none;  
proc surveymeans data=ech_strata N=210 sum;  
    var naissance;  
    strata strate / list;  
    by replicate;  
    weight SamplingWeight;  
    ods output Statistics=estimations;  
run;  
ods select all;
```

```
data estimations;  
    set estimations;  
    err=sum-2624;  
    err_rel=err/2624;  
    err2=err**2;
```

```
run;
```

```
proc means data=estimations mean;
```

```
var err_rel err2;
```

```
run;
```

Tableau des résultats		
	$BR_{MC}$	$V_{MC}$
Répartition de Neyman		
Répartition proportionnelle		

### 5.3 Exercice – Comparaison de l'EASSR et du plan stratifié

Cet exercice permettra de **comparer l'efficacité du plan de sondage stratifié** (avec répartition proportionnelle et EASSR dans les strates) **avec celle de l'EASSR**. Nous effectuerons une **étude par simulation** à partir de la base de données des paroisses afin d'approximer la variance de l'estimateur du total des naissances. Pour ce faire, nous tirerons 1000 échantillons répliqués de taille  $n = 50$ . Vous avez vu que

$$V_{EASSR}(\hat{t}_y) - V_{PROP}(\hat{t}_{st}) \geq 0 \Leftrightarrow SSB \geq \sum_{h=1}^H \left(1 - \frac{N_h}{N}\right) S_h^2.$$

Afin d'observer l'effet de l'**homogénéité dans les strates** sur la différence d'efficacité entre les deux plans de sondage, nous utiliserons d'une part, la variable « strate » (construite à partir de « pop99 ») pour indiquer les strates; d'autre part, la variable « region ».

/\*On calcule la variance Monte Carlo pour un EASSR\*/

```
proc surveysselect data=pop_paroisses method=srs n=50 rep=1000 out=ech_srs stats;
```

```
run;
```

```

proc means data=ech_srs noprint;
    var naissance;
    by replicate;
    weight SamplingWeight;
    output out=estimations sum=tot_y_estime;
run;

```

```

data estimations;
    set estimations;
    err=tot_y_estime-2624;
    err2=err**2;
run;

```

```

proc means data=estimations mean;
    var err2;
run;

```

*/\*On calcule la variance Monte Carlo pour l'échantillonnage stratifié\*/*

```

%repartition(pop=pop_paroisses,type=prop,idstrate=strate,interet=naissance,n=50);

```

```

proc surveyselect data=pop_paroisses method=srs samsize=work.allocation2 rep=1000
out=ech_strata;
    strata strate;
run;

```

```

proc sort data=ech_strata;
    by replicate strate;
run;

```

```

ods select none;
proc surveymeans data=ech_strata N=210 sum;
    var naissance;

```

```

strata strate / list;
by replicate;
weight SamplingWeight;
ods output Statistics=estimations;

run;

ods select all;

data estimations;
set estimations;
err=sum-2624;
err2=err**2;

run;

proc means data=estimations mean;
var err2;

run;

```

Tableau des résultats		
	$V_{MC}$ EASSR	$V_{MC}$ stratifié
Id strate: strate		
Id strate: region		

## Fiche 6

Dans cette fiche, nous verrons comment utiliser SAS pour de l'estimation par domaine et pour l'estimation d'un ratio. Un exercice sera également présenté afin d'illustrer le problème du biais de l'estimateur d'un ratio lorsque la taille de l'échantillon est petite.

### 6.1 Estimation du total d'un domaine

L'estimation du total d'un domaine peut facilement être obtenue en SAS avec la procédure SURVEYMEANS. Il suffit d'ajouter l'**option DOMAIN** en spécifiant la variable indicatrice du domaine. La procédure SURVEYMEANS utilise la formule générale suivante pour

estimer  $t_d = \sum_{i \in U} \delta_i y_i$  :

$$\hat{t}_d = \sum_{i \in s} \delta_i \frac{y_i}{\pi_i}.$$

Dans le cas d'un EASSR, cette formule se simplifie pour donner celle vue en classe. De plus, l'estimation de la variance de  $\hat{t}_d$  est donnée par :

$$\hat{V}(\hat{t}_d) = \left(1 - \frac{n}{N}\right) \frac{n}{n-1} \sum_{i \in s} \left( \delta_i \frac{y_i}{\pi_i} - \frac{\sum_{i \in s} \delta_i \frac{y_i}{\pi_i}}{n} \right)^2,$$

qui se simplifie dans le cas de l'EASSR pour donner

$$\hat{V}(\hat{t}_d) = N^2 \left(1 - \frac{n}{N}\right) \frac{s_u^2}{n},$$

où  $s_u^2 = \frac{1}{n-1} \sum_{i \in s} \left( \delta_i y_i - \frac{\sum_{i \in s} \delta_i y_i}{n} \right)^2$ . On peut montrer (preuve laissée en exercice) que cette

expression est équivalente à l'expression (4.6) des notes de cours (p. 75).

À partir de la base de sondage des professeurs, nous estimerons le total du salaire en 2001 pour chaque sexe de même que le total d'hommes et de femmes dans la population.

`libname mylib 'stt2000/tp6';`

```

data pop_professeurs;
    set mylib.pop_professeurs;
    one=1;
run;

title "Vrais totaux";
proc means data=pop_professeurs sum;
    var sal01 one;
    class sexe;
run;

proc surveyselect data=pop_professeurs method=srs n=30 stats out=ech_srs;
run;

title "Estimation du total d'un domaine";
proc surveymeans data=ech_srs N=200 sum;
    var sal01 one;
    domain sexe;
    weight SamplingWeight;
    ods output Domain=estimations;
run;

```

Vrais totaux				
The MEANS Procedure				
sexe	N Obs	Variable	Sum	
0	81	sal01	3676869.00	
		one	81.0000000	
1	119	sal01	6130112.00	
		one	119.0000000	

Estimation du total d'un domaine			
The SURVEYMEANS Procedure			
Data Summary			
Number of Observations		30	
Sum of Weights		200	
Statistics			
Variable	Sum	Std Dev	
sal01	9072873	339678	
one	200.000000	0.000000447	
Domain Analysis: sexe			
sexe	Variable	Sum	Std Dev
0	sal01	4450913	750501
	one	106.666667	17.082177
1	sal01	4621960	870673
	one	93.333333	17.082177

## 6.2 Estimation d'un ratio

a)

L'estimation d'un ratio peut facilement être obtenue en SAS avec la procédure SURVEYMEANS. Il suffit de spécifier l'**option RATIO** qui prend comme paramètre les variables au numérateur, puis au dénominateur. Ainsi, la procédure calcule l'estimateur du ratio  $B = t_{y_1} / t_{y_2}$  avec

$$\hat{B} = \frac{\sum_{i \in s} y_{1i} / \pi_i}{\sum_{i \in s} y_{2i} / \pi_i},$$

qui se simplifie dans le cas d'un EASSR pour donner l'estimateur que vous connaissez:

$$\hat{B} = \frac{\sum_{i \in s} y_{1i}}{\sum_{i \in s} y_{2i}}$$

De plus, l'estimation de la variance d'un ratio est calculée selon une expression (pas rapportée ici) se simplifiant dans le cas de l'EASSR pour donner l'expression (4.11) des notes de cours (p. 80). Cette approximation est valide si la taille échantillonnale  $n$  est suffisamment grande.

```
proc surveystat data=pop_professeurs method=srs n=30 stats out=ech_srs;
run;
```

```
title "Estimation d'un ratio";
proc surveymeans data=ech_srs N=200 sum varsum;
    ratio sal01/saldebut;
    weight SamplingWeight;
    ods output Ratio=estimations;
run;
```

Estimation d'un ratio				
The SURVEYMEANS Procedure				
Data Summary				
Number of Observations		30		
Sum of Weights		200		
Statistics				
Variable	Label	Sum	Std Dev	Var of Sum
id		23580	2053.487026	4216809
sexe		153.333333	14.482120	209.731801
datentree		396287	282.587524	79856
dep		886.666667	68.300104	4664.904215
saldebut		2117260	196260	38518100876
sal01		9792100	301670	91004751236
exp		5866.666667	193.962512	37621
anc		3913.333333	282.587524	79856
SelectionProb	Probability of Selection	30.000000	0	0
Ratio Analysis				
	Numerator	Denominator	Ratio	Std Err
	sal01	saldebut	4.624893	0.554652

b)

Vous avez vu en classe comment effectuer une **linéarisation du premier ordre** de l'estimateur du ratio. Vous avez obtenu:

$$\hat{B} - B \approx \hat{t}_z,$$

où

$$z_i = \frac{y_{1i} - B y_{2i}}{t_{y2}}.$$

Ainsi,  $Biais(\hat{B}) \approx 0$  et  $V(\hat{B}) \approx V(\hat{t}_z)$  si la taille échantillonnale  $n$  est suffisamment grande. Une autre façon d'obtenir l'approximation du premier ordre de la variance estimée de  $\hat{B}$  en SAS serait donc d'utiliser la procédure SURVEYMEANS sur la **variable linéarisée**

$$\hat{z}_i = \frac{y_{1i} - \hat{B} y_{2i}}{\hat{t}_{y2}}.$$

```
proc means data=ech_srs sum noprint;
    var sal01 saldebut;
    output out=totaux_ech sum=total_1_ech total_2_ech;
run;
```

```
data totaux_ech;
    set totaux_ech;
    B_estime=total_1_ech/total_2_ech;
    total_2_estime=total_2_ech*200/30;
    call symput('B_estime',B_estime);
    call symput('total_2_estime',total_2_estime);
run;
```

```
data ech_srs;
    set ech_srs;
    z_estime=(sal01-&B_estime*saldebut)/&total_2_estime;
run;
```

```
proc surveymeans data=ech_srs N=200 sum varsum;
```

```

var z_estime;
weight SamplingWeight;
ods output Statistics=estimations;
run;

```

Estimation d'un ratio			
The SURVEYMEANS Procedure			
Data Summary			
	Number of Observations		30
	Sum of Weights		200
Statistics			
Variable	Sum	Std Dev	Var of Sum
z_estime	4.936333E-12	0.554652	0.307639

### 6.3 Estimation de la moyenne d'un domaine

a)

L'estimation de la moyenne d'un domaine peut facilement être obtenue en SAS avec la procédure SURVEYMEANS. Il suffit encore une fois d'ajouter l'**option DOMAIN** en spécifiant la variable indicatrice du domaine. Afin d'estimer la moyenne d'un domaine,  $\bar{y}_{U_d}$ , la procédure SURVEYMEANS utilise l'expression

$$\bar{y}_d = \frac{\sum_{i \in s} \delta_i \frac{y_i}{\pi_i}}{\sum_{i \in s} \delta_i \frac{1}{\pi_i}},$$

qui se simplifie, dans le cas de l'EASSR, pour donner la formule vue en classe:

$$\bar{y}_d = \frac{\sum_{i \in s} \delta_i y_i}{\sum_{i \in s} \delta_i}.$$

De plus, l'estimation de la variance de  $\bar{y}_d$  est calculée selon une expression (pas rapportée ici) se simplifiant dans le cas de l'EASSR pour donner :

$$\hat{V}(\bar{y}_d) \approx \left(1 - \frac{n}{N}\right) \frac{1}{n} \left(\frac{1}{p_d}\right)^2 \left(\frac{n_d - 1}{n - 1}\right) s_{yd}^2,$$

qui est une expression plus précise que l'expression (4.17) des notes de cours (p. 83).

```
proc surveysselect data=pop_professeurs method=srs n=30 stats out=ech_srs;
run;
```

```
title "Estimation de la moyenne d'un domaine";
proc surveymeans data=ech_srs N=200 mean var;
    var sal01;
    domain sexe;
    weight SamplingWeight;
    ods output Domain=estimations;
run;
```

Estimation de la moyenne d'un domaine				
The SURVEYMEANS Procedure				
Data Summary				
Number of Observations		30		
Sum of Weights		200		
Statistics				
Variable	Mean	Std Error of Mean	Var of Mean	
sal01	49418	1329.230423	1766854	
Domain Analysis: sexe				
sexe	Variable	Mean	Std Error of Mean	Var of Mean
0	sal01	47025	1984.372195	3937733
1	sal01	51248	1673.274482	2799847

b)

Vous avez vu qu'estimer la moyenne d'un domaine est équivalent à estimer le ratio de deux totaux,  $u_i = \delta_i y_i$  et  $\delta_i$ . On peut donc également obtenir l'approximation du premier ordre de l'estimateur de la variance de  $\bar{y}_d$  en SAS en utilisant la procédure SURVEYMEANS sur une **variable linéarisée**  $\hat{z}_i$ , similairement à ce qui a été fait en 6.2 b). Ainsi,  $\hat{z}_i$  est donnée par:

$$\hat{z}_i = \frac{u_i - \bar{y}_d \delta_i}{\hat{N}_d}.$$

Le code qui suit permet d'estimer la variance de  $\bar{y}_d$  lorsque le domaine est donné par sexe = 1, ce qui représente le sexe masculin.

```
data ech_srs;
    set ech_srs;
    u=sexe*sal01;
run;

proc means data=ech_srs sum noprint;
    var u sexe;
    output out=totaux_ech sum=total_1_ech total_2_ech;
run;

data totaux_ech;
    set totaux_ech;
    B_estime=total_1_ech/total_2_ech;
    total_2_estime=total_2_ech*200/30;
    call symput('B_estime',B_estime);
    call symput('total_2_estime',total_2_estime);
run;

data ech_srs;
    set ech_srs;
    l=(u-&B_estime*sexe)/&total_2_estime;
run;
```

```

proc surveymeans data=ech_srs N=200 sum varsum;
    var l;
    weight SamplingWeight;
    ods output Statistics=estimations;
run;
title;

```

Estimation de la moyenne d'un domaine			
The SURVEYMEANS Procedure			
Data Summary			
Number of Observations			30
Sum of Weights			200
Statistics			
Variable	Sum	Std Dev	Var of Sum
1	-0.000000471	1673.274483	2799847

#### 6.4 Exercice – Effet d'une petite taille d'échantillon sur le biais de l'estimateur d'un ratio

Avec un développement en série de Taylor du premier ordre, vous avez vu que **si la taille échantillonnale  $n$  est suffisamment grande**, alors l'estimateur du ratio  $\hat{B}$  est approximativement sans biais pour le vrai ratio  $B$ . Dans cet exercice, nous observerons l'effet d'une **petite taille échantillonnale** sur le **biais de  $\hat{B}$** . À cette fin, nous effectuerons une étude Monte Carlo à partir de la base de sondage des professeurs. Nous nous intéresserons au ratio du total de la variable « sal01 » (salaire en 2001) par rapport au total de la variable « saldebut » (salaire à l'embauche). Nous calculerons le biais relatif Monte Carlo à partir de  $R=1000$  estimations du ratio pour différentes tailles  $n$  : 2, 10, 30, 100.

```
/*On calcule le vrai ratio B*/
```

```
proc means data= pop_professeurs sum noprint;  
    var sal01 saldebut;  
    output out=totaux_pop sum=total_1_pop total_2_pop;  
run;
```

```
data totaux_pop;  
    set totaux_pop;  
    B=total_1_pop/total_2_pop;  
    call symput('B',B);  
run;
```

```
/*On tire R=1000 échantillons répliqués par EASSR*/
```

```
proc surveysselect data=pop_professeurs method=srs n=2 rep=1000 stats out=ech_srs;  
run;
```

```
/* On calcule l'estimateur du total pour « sal01 » et « saldebut »*/
```

```
proc means data=ech_srs sum noprint;  
    var sal01 saldebut;  
    by replicate;  
    output out=totaux_ech sum=total_1_ech total_2_ech;  
run;
```

```
/* On calcule l'erreur relative pour les R=1000 répliqués*/
```

```
data totaux_ech;  
    set totaux_ech;  
    B_estime=total_1_ech/total_2_ech;  
    err=B_estime-&B;  
    err_rel=err/&B*100;
```

```
run;
```

```
/*On calcule le biais relatif Monte Carlo*/
```

```
proc means data=totaux_ech mean;
```

```
var err_rel;
```

```
run;
```

Tableau des résultats	
	$BR_{MC}$
$n = 2$	%
$n = 10$	%
$n = 30$	%
$n = 100$	%

## Fiche 7

Dans cette fiche, nous introduirons d'abord la procédure REG qui sert à ajuster des modèles de régression linéaire. Puis, nous montrerons comment calculer, en SAS, les estimateurs à utiliser en présence d'information auxiliaire : estimateur post-stratifié, par le ratio, par la régression et par la différence. Enfin, un exercice permettra de comparer les trois derniers estimateurs pour différentes populations.

### 7.1 La procédure REG

La procédure REG permet d'estimer les paramètres  $B_0$  et  $B_1$  dans un modèle de régression linéaire simple :

$$y_i = B_0 + B_1 x_i + e_i.$$

**L'option OUTEST=** permet d'écrire les paramètres estimés,  $\hat{B}_0$  et  $\hat{B}_1$ , dans une table SAS.

La procédure REG permet aussi, par défaut, de tester

$$H_0: B_0 = 0 \\ \text{contre } H_1: B_0 \neq 0$$

en regardant la valeur- $p$  de  $B_0$  (intercept) dans le output. Cela sera utile pour justifier l'utilisation de l'estimateur par le ratio. Aussi, en ajoutant **l'instruction TEST**, on peut tester d'autres hypothèses comme

$$H_0: B_1 = 1 \\ \text{contre } H_1: B_1 \neq 1,$$

ce qui sera utile pour justifier l'utilisation de l'estimateur par la différence. Les valeurs prédites,

$$\hat{y}_i = \hat{B}_0 + \hat{B}_1 x_i,$$

et les résidus estimés,  $y_i - \hat{y}_i$ , pour chaque unité  $i$  peuvent également être écrits dans une table SAS en utilisant l'**instruction OUTPUT**. Cela pourrait être utile si on voulait calculer l'estimateur de la variance pour les estimateurs par le ratio ou par la régression.

La procédure REG permet aussi, avec l'**instruction PLOT**, de produire un graphique sur lequel figurent les observations et la droite de régression estimée. Enfin, notons que la procédure REG en est une **interactive**, c'est-à-dire qu'elle ne s'arrête pas après l'instruction RUN. Ainsi, elle permet, par exemple, de modifier le modèle sans avoir à relancer un nouveau PROC REG. Pour terminer la procédure, il faut donc ajouter l'**instruction QUIT**.

```
libname mylib "st2000/tp7/ ";
```

```
data pop1;
```

```
    set mylib.pop1;
```

```
run;
```

```
title "Modele de regression";
```

```
proc reg data=pop1 outest=parametres;
```

```
    model y=x;
```

```
    test x=1;
```

```
    plot y*x;
```

```
    output out=predictions predicted=yhat residual=yresid;
```

```
run;
```

```
quit;
```

```
title;
```

```

                                     Modele de regression
                                     The REG Procedure
                                     Model: MODEL1
                                     Dependent Variable: y
Number of Observations Read          500
Number of Observations Used          500
```

### Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	1937910	1937910	849.89	<.0001
Error	498	1135533	2280.18699		
Corrected Total	499	3073443			

Root MSE	47.75130	R-Square	0.6305
Dependent Mean	201.42725	Adj R-Sq	0.6298
Coeff Var	23.70648		

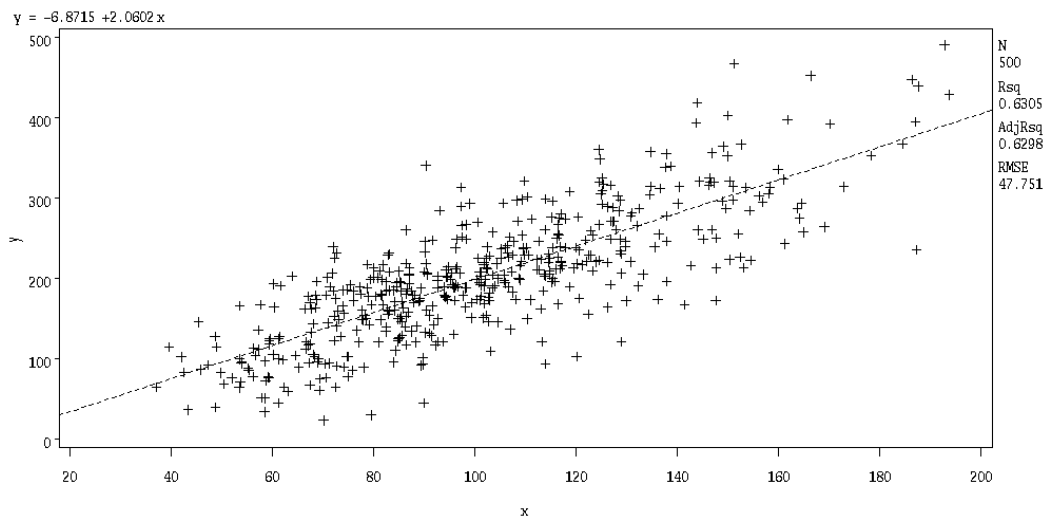
### Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr >  t
Intercept	1	-6.87154	7.45735	-0.92	0.3573
x	1	2.06021	0.07067	29.15	<.0001

### Test 1 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	1	513208	225.07	<.0001
Denominator	498	2280.18699		

### Modele de regression



## 7.2 Estimation en présence d'information auxiliaire

Dans cette section, nous verrons comment calculer les différents estimateurs utilisés en présence d'information auxiliaire: l'estimateur post-stratifié, par le ratio, par la régression et par la différence.

a)

Nous traiterons le cas de l'**estimateur post-stratifié** d'une moyenne en tirant un échantillon EASSR de taille  $n = 30$  de la population des professeurs. Nous utiliserons la variable « sexe » pour indiquer les post-strates. Nous utiliserons la formule suivante pour calculer l'estimateur :

$$\bar{y}_{post} = \frac{1}{N} \sum_{j=1}^J \sum_{i \in s_j} \frac{N_j}{n_j} y_i.$$

```
data pop_professeurs;
    set mylib.pop_professeurs(keep=id sexe sal01);
run;

/*On calcule les  $N_j$  à partir de la population*/

proc sort data=pop_professeurs;
    by sexe;
run;

proc means data=pop_professeurs noprint;
    var id;
    by sexe;
    output out=tailles_pop(drop=_TYPE_ _FREQ_) N=nj_pop;
run;

/*On tire l'échantillon*/

proc surveyselect data=pop_professeurs method=srs n=30 stats out=ech_srs noprint;
```

```

run;

/*On calcule les  $n_j$  à partir de l'échantillon*/

proc means data=ech_srs noprint;
    var id;
    by sexe;
    output out=tailles_ech(drop=_TYPE_ _FREQ_) N=nj_ech;
run;

/*On calcule les nouveaux poids*/

data ech_srs;
    merge ech_srs tailles_ech tailles_pop;
    by sexe;
    w=nj_pop/nj_ech;
run;

/*On calcule l'estimateur post-stratifié*/

proc surveymeans data=ech_srs N=200 mean;
    var sal01;
    weight w;
run;

```

b)

Nous traiterons le cas de **l'estimateur par le ratio** d'une moyenne en tirant un échantillon EASSR de taille  $n = 30$  de la population 1. Nous utiliserons la formule suivante pour calculer l'estimateur :

$$\hat{y}_r = \hat{B}\bar{x}_U,$$

où  $\hat{B} = \frac{\hat{t}_y}{\hat{t}_x}$ .

```

data pop1;
    set mylib.pop1;
run;

/*On calcule la moyenne des x dans la population*/

proc means data=pop1 mean noprint;
    var x;
    output out=moy_x_pop(drop=_TYPE_) mean=moy_x_pop;
run;

/*On tire l'échantillon*/

proc surveyselect data=pop1 method=srs n=30 stats out=ech_srs;
run;

/*On estime B*/

proc surveymeans data=ech_srs N=500 mean;
    ratio y/x;
    weight SamplingWeight;
    ods output Ratio=B_estime(keep=ratio);
run;

/*On calcule l'estimateur par le ratio de la moyenne*/

data estimations;
    merge moy_x_pop B_estime;
    moy_y_estimee=ratio*moy_x_pop;
run;

```

c)

Nous traiterons le cas de **l'estimateur par la régression** d'une moyenne. Nous utiliserons le même échantillon qu'en b). Nous utiliserons la formule suivante pour calculer l'estimateur :

$$\hat{y}_{reg} = \hat{B}_0 + \hat{B}_1 \bar{x}_U.$$

*/\*On estime  $B_0$  et  $B_1$  avec un modèle de régression linéaire simple\*/*

```
proc reg data=ech_srs outest=parametres;  
    model y=x;  
run;  
quit;
```

*/\*On calcule l'estimateur par la régression de la moyenne\*/*

```
data estimations;  
    merge moy_x_pop parametres;  
    moy_y_estimee=intercept+x* moy_x_pop;  
    keep moy_y_estimee;  
run;
```

d)

Nous traiterons le cas de **l'estimateur par la différence** d'une moyenne. Nous utiliserons le même échantillon qu'en b). Nous utiliserons la formule suivante pour calculer l'estimateur :

$$\hat{y}_{diff} = \bar{x}_U + \bar{d}_s,$$

$$\text{où } \bar{d}_s = \frac{1}{n} \sum_{i \in s} d_i = \frac{1}{n} \sum_{i \in s} (y_i - x_i).$$

*/\*On calcule les  $d_i$ \*/*

```
data ech_srs;  
    set ech_srs;
```

```

d=y-x;
run;

/*On calcule la moyenne des  $d_i$  dans l'échantillon*/

proc means data=ech_srs mean noprint;
var d;
output out=moy_d_ech mean=moy_d_ech;
run;

/*On calcule l'estimateur par la différence de la moyenne*/

data estimations;
merge moy_x_pop moy_d_ech;
moy_y_estimee= moy_x_pop+moy_d_ech;
keep moy_y_estimee;
run;

```

### 7.3 Exercice – Comparaison des estimateurs par le ratio, la regression et la différence

Cet exercice a pour but de comparer les estimateurs par le ratio, par la régression et par la différence et d'observer dans quelle situation ils sont appropriés. Pour ce faire, nous avons généré quatre populations de taille  $N = 500$  à partir des modèles figurant dans le tableau ci-dessous. Pour les trois premières populations,  $e_i$ , qui représente le terme d'erreur, suit une loi normale centrée à zéro et de variance telle que  $R^2 = 0.64$ .

	Modèle
pop1	$y_i = 2x_i + e_i$
pop2	$y_i = x_i + e_i$
pop3	$y_i = 500 + 2x_i + e_i$
pop4	Pas de relation entre $x$ et $y$

Nous allons effectuer une **étude par simulation de Monte Carlo**. Nous tirerons d'abord  $R = 1000$  échantillons EASSR de  $n = 30$  unités, puis nous calculerons le biais Monte Carlo et l'erreur quadratique moyenne Monte Carlo pour chaque estimateur de la moyenne. Nous nous intéresserons également à l'estimateur de la moyenne de type Horvitz-Thompson,  $\bar{y}_s$ , à titre comparatif.

```
data pop1;
```

```
    set mylib.pop1;
```

```
run;
```

```
/*On effectue l'analyse de régression à partir de la population*/
```

```
proc reg data=pop1;
```

```
    model y=x;
```

```
    test x=1;
```

```
run;
```

```
quit;
```

```
/*On calcule la moyenne des x et des y dans la population*/
```

```
proc means data=pop1 mean noprint;
```

```
    var x y;
```

```
    output out=moy_xy_pop(drop=_TYPE_) mean=moy_x_pop moy_y_pop;
```

```
run;
```

```
data _NULL_;
```

```
    set moy_xy_pop;
```

```
    call symput('moy_x_pop', moy_x_pop);
```

```
    call symput('moy_y_pop', moy_y_pop);
```

```
run;
```

```
/*On tire les échantillons*/
```

```
proc surveyselect data=pop1 method=srs n=30 rep=1000 stats out=ech_srs;  
run;
```

```
/*On calcule les estimateurs par le ratio*/
```

```
ods select none;  
proc surveymeans data=ech_srs N=500 mean;  
    ratio y/x;  
    weight SamplingWeight;  
    by replicate;  
    ods output Ratio=B_estime(keep=ratio);  
run;  
ods select all;
```

```
data estimations_ratio;  
    set B_estime;  
    moy_y_estimee_ratio=ratio*&moy_x_pop;  
    keep moy_y_estimee_ratio;  
run;
```

```
/*On calcule les estimateurs par la régression*/
```

```
proc reg data=ech_srs outest=parametres noprint;  
    model y=x;  
    by replicate;  
run;  
quit;
```

```
data estimations_reg;  
    set parametres;  
    moy_y_estimee_reg=intercept+x*&moy_x_pop;  
    keep moy_y_estimee_reg;  
run;
```

```
/*On calcule les estimateurs par la différence*/
```

```
data ech_srs;  
    set ech_srs;  
    d=y-x;  
run;
```

```
proc means data=ech_srs mean noprint;  
    var d;  
    by replicate;  
    output out=moy_d_ech mean=moy_d_ech;  
run;
```

```
data estimations_diff;  
    set moy_d_ech;  
    moy_y_estimee_diff= &moy_x_pop+moy_d_ech;  
    keep moy_y_estimee_diff;  
run;
```

```
/*On calcule les estimateurs de type Horvitz-Thompson*/
```

```
proc means data=ech_srs noprint;  
    var y;  
    by replicate;  
    weight SamplingWeight;  
    output out=estimations_HT mean=moy_y_estimee_HT;  
run;
```

```
/*On calcule les mesures Monte Carlo (biais et EQM)*/
```

```
data estimations;  
    merge estimations_HT estimations_ratio estimations_reg estimations_diff;  
    err_HT=moy_y_estimee_HT-&moy_y_pop;  
    err_ratio=moy_y_estimee_ratio-&moy_y_pop;
```

```

err_reg=moy_y_estimee_reg-&moy_y_pop;
err_diff=moy_y_estimee_diff-&moy_y_pop;
err2_HT=err_HT**2;
err2_ratio=err_ratio**2;
err2_reg=err_reg**2;
err2_diff=err_diff**2;

run;

proc means data=estimations mean;
run;

```

Tableau d'étude des populations				
	Modèle	$R^2$	$B_0 = 0 ?$	$B_1 = 1 ?$
pop1				
pop2				
pop3				
pop4				

Tableau des résultats (pop1)		
	$Biais_{MC}$	$EQM_{MC}$
Ratio		
Rég.		
Diff.		
HT		

Tableau des résultats (pop3)		
	$Biais_{MC}$	$EQM_{MC}$
Ratio		
Rég.		
Diff.		
HT		

Tableau des résultats (pop2)		
	$Biais_{MC}$	$EQM_{MC}$
Ratio		
Rég.		
Diff.		
HT		

Tableau des résultats (pop4)		
	$Biais_{MC}$	$EQM_{MC}$
Ratio		
Rég.		
Diff.		
HT		

## ANNEXE A - Les opérateurs en SAS

### Opérateurs arithmétiques

<b>Symbole</b>	<b>Signification</b>
+	Addition
-	Soustraction
*	Multiplication
/	Division
**	Exponentiation
><	Maximum
<>	Minimum

### Opérateurs de comparaison

<b>Symbole</b>	<b>Signification</b>
= ou EQ	Égal à
<> ou NE	Différent de
> ou GT	Supérieur à
< ou LT	Inférieur à
>= ou GE	Supérieur ou égal à
<= ou LE	Inférieur ou égal à

### Opérateurs logiques

<b>Symbole</b>	<b>Signification</b>
AND	Et
OR	Ou
NOT	Non

### Opérateurs sur les variables caractères

<b>Symbole</b>	<b>Signification</b>
	Concaténation

## **ANNEXE B - Les études par simulation en théorie de l'échantillonnage**

### Introduction

La simulation (ou simulation Monte Carlo) est une technique numérique qui permet d'effectuer des expériences sur un ordinateur. En statistique, ces expériences ont pour but d'étudier les propriétés de méthodes statistiques.

L'étude par simulation, à l'instar de toute autre expérience statistique, requiert une planification minutieuse. La formulation de l'énoncé des objectifs est une étape importante dans l'élaboration d'une étude par simulation. En effet, avant d'entreprendre la mise en œuvre d'aspects plus techniques (l'aspect algorithmique pour n'en citer qu'un seul), il faut pouvoir répondre le plus clairement possible à des questions telles que : Que cherche-t-on à démontrer? Quelles méthodes cherche-t-on à comparer? Quelles données utilisera-t-on?

Dans le cadre de ce cours, les études par simulation seront utilisées afin d'étudier l'effet de l'échantillonnage. Pour ce faire, le mécanisme aléatoire de la sélection de l'échantillon devra être répété un grand nombre de fois. Les études par simulation nous permettront entre autres d'étudier les propriétés des estimateurs présentés en classe et de comparer ces derniers entre eux.

### Définir la population de départ

En premier lieu, il faut se donner une population qui servira de point de départ. En pratique, deux types de populations sont généralement utilisés :

(i) La population est construite à partir de l'échantillon observé à un moment donné dans une enquête donnée. La population peut, par exemple, consister au choix des répondants complets seulement (autrement dit, on ne conserve que les unités qui ont répondu à toutes les variables de l'enquête et qui ont satisfait toutes les règles de vérification).

(ii) La population peut être générée à partir de certains modèles statistiques. Par exemple, on peut construire une population de taille  $N = 1000$  contenant deux variables ( $x$  et  $y$ ). D'abord, 1000 réalisations de la variable  $x$  sont générées selon une distribution donnée (par exemple, selon une loi exponentielle de paramètre  $\alpha$ ). Ensuite, étant données les réalisations de la variable  $x$ , on génère 1000 réalisations de la variable  $y$  selon le modèle

$$y = \beta x + \varepsilon,$$

où les erreurs  $\varepsilon$  sont générées selon une distribution donnée (par exemple, selon une loi normale de moyenne 0 et de variance  $\sigma^2$ ). Il faudra donc fixer les valeurs des paramètres  $\alpha$ ,  $\beta$  et  $\sigma^2$ . Le choix de ces valeurs permettra de contrôler plusieurs caractéristiques de la population telles que la moyenne des variables  $x$  et  $y$ , leur variabilité, leur coefficient d'asymétrie, ou encore le coefficient de corrélation entre  $x$  et  $y$ .

Un avantage important d'une population du type (i) est que celle-ci consiste en des données observées et donc réelles, ce qui assure que le contexte dans lequel nous effectuons les simulations est similaire à celui qui prévaut en pratique. Cependant, les modèles générant la population ne sont pas connus; de plus, nous sommes contraints à travailler avec les caractéristiques (moyenne et variabilité des variables d'intérêt ou coefficient de corrélation entre deux variables) de la population choisie. Bien que les données ne soient pas réelles dans le cas d'une population de type (ii), les modèles la générant sont parfaitement contrôlés, ce qui facilite l'étude des propriétés d'une méthode statistique ainsi que l'interprétation des résultats. De plus, il est possible faire varier certaines caractéristiques de la population (par exemple, le coefficient de corrélation entre deux variables) et d'étudier l'effet de ces variations sur les résultats.

### Étapes d'une étude par simulation

Supposons qu'une population  $U$  de taille  $N$  est disponible. L'objectif est d'estimer un paramètre  $\theta$  de cette population. Habituellement,  $\theta$  désigne le total ou la moyenne d'une variable d'intérêt, le total ou la moyenne d'un domaine, un quantile, un coefficient de régression ou de corrélation, etc. Soit  $\hat{\theta}$  un estimateur de  $\theta$ . Rappelons que le mécanisme d'échantillonnage qui conduit à l'échantillon  $s$  sert de point de départ pour étudier le comportement (par exemple, biais et variance) des estimateurs.

Rappelons que l'espérance de  $\hat{\theta}$ , que l'on désigne par  $E(\hat{\theta})$  est définie selon

$$E(\hat{\theta}) = \sum_{s \in \Omega} \hat{\theta} p(s), \quad (\text{B.1})$$

où  $\Omega$  désigne l'ensemble de tous les échantillons possibles et  $p(s)$  désigne la probabilité de tirer l'échantillon  $s$ . De plus, la variance de  $\hat{\theta}$ , que l'on désigne par  $V(\hat{\theta})$  est définie selon

$$E\left(\hat{\theta} - E(\hat{\theta})\right)^2 = \sum_{s \in \Omega} \left(\hat{\theta} - E(\hat{\theta})\right)^2 p(s) \quad (\text{B.2})$$

et l'erreur quadratique moyenne de  $\hat{\theta}$ , que l'on désigne par  $EQM(\hat{\theta})$ , est définie selon

$$E(\hat{\theta} - \theta)^2 = \sum_{s \in \Omega} (\hat{\theta} - \theta)^2 p(s). \quad (\text{B.3})$$

Notons que dans le contexte d'une étude par simulation, les paramètres de la population sont connus. On connaît donc la vraie valeur du paramètre  $\theta$  et, dans certains cas, on connaît la vraie valeur de  $V(\hat{\theta})$ .

On procédera selon les grandes étapes suivantes :

- (1) Tirer un échantillon aléatoire  $s$ , de taille  $n$ , selon un plan de sondage donné.
- (2) Calculer l'estimateur  $\hat{\theta}$  (et tout autre estimateur d'intérêt).
- (3) Répéter les étapes (1)-(2)  $R$  fois.
- (4) Calculer les mesures Monte Carlo appropriées.

Soit  $\hat{\theta}_j$  l'estimateur  $\hat{\theta}$  pour le  $j^{\text{e}}$  échantillon,  $j=1, \dots, R$ . Il est important de noter que les variables aléatoires,  $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_R$  sont indépendantes et identiquement distribuées de moyenne  $E(\hat{\theta})$  et de variance  $V(\hat{\theta})$ . Nous décrivons maintenant chacune des étapes (1)-(4) en détail :

(1) Il faut choisir un plan de sondage (sondage aléatoire simple sans remise, échantillonnage stratifié, échantillonnage à plusieurs degrés, etc.). Il faut également choisir la (les) valeur (s) de  $n$  ou la (les) valeur (s) de la fraction de sondage  $n/N$ . Notons que le logiciel SAS offre la procédure SURVEYSELECT qui permet de tirer facilement des échantillons selon de nombreux plans de sondage (sondage aléatoire simple avec et sans remise, échantillonnage stratifié, échantillonnage proportionnel à la taille etc.). De plus, l'option REP permet de tirer  $R$  échantillons, ce qui s'avère très utile dans le contexte d'une étude par simulation.

(2) Dans l'univers des sondages, on s'intéresse habituellement à illustrer les propriétés de certains estimateurs ponctuels de paramètres univariés tels qu'une moyenne ou un total dans la population ou de paramètres bivariés tels qu'un coefficient de régression ou de corrélation. On s'intéresse également à des estimateurs de variance.

(3) Une étape importante dans toute étude par simulation est le choix du nombre d'itérations  $R$ . Nous en discutons en section 4.

(4) Soient  $\hat{\theta}$  un estimateur  $\theta$  et  $\hat{V}(\hat{\theta})$  un estimateur de  $V(\hat{\theta})$ . L'espérance de  $\hat{\theta}$ ,  $E(\hat{\theta})$ , définie par (B.1) peut être approximée par la moyenne observée (ou l'espérance Monte Carlo) des  $R$  valeurs  $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_R$  donnée par

$$E_{MC}(\hat{\theta}) \equiv \hat{\theta}_{MC} = \frac{1}{R} \sum_{j=1}^R \hat{\theta}_j. \quad (\text{B.4})$$

La loi des grands nombres nous garantit que pour toute grande valeur de  $R$ , l'espérance Monte Carlo,  $E_{MC}(\hat{\theta})$ , est très voisine de  $E(\hat{\theta})$ . Il existe de nombreuses mesures Monte Carlo utilisées dans les études par simulation. Dans ce qui suit, nous décrivons quelques unes fréquemment utilisées :

(a) Le biais relatif (en %) de  $\hat{\theta}$ , défini par  $BR(\hat{\theta}) = \frac{E(\hat{\theta}) - \theta}{\theta} \times 100$ , peut être approximé par le biais relatif Monte Carlo (en %) donné par

$$BR_{MC}(\hat{\theta}) = \frac{E_{MC}(\hat{\theta}) - \theta}{\theta} \times 100. \quad (\text{B.5})$$

(b) La variance de  $\hat{\theta}$ , définie en (B.2) peut être approximée par la variance Monte Carlo donnée par

$$V_{MC}(\hat{\theta}) = \frac{1}{R} \sum_{j=1}^R [\hat{\theta}_j - E_{MC}(\hat{\theta})]^2. \quad (\text{B.6})$$

(c) L'erreur quadratique moyenne de  $\hat{\theta}$  en (B.3) peut être approximée par l'erreur quadratique moyenne Monte Carlo donnée par

$$EQM_{MC}(\hat{\theta}) = \frac{1}{R} \sum_{j=1}^R (\hat{\theta}_j - \theta)^2. \quad (\text{B.7})$$

Lorsque l'estimateur  $\hat{\theta}$  est sans biais pour  $\theta$ , on a  $EQM_{MC}(\hat{\theta}) \approx V_{MC}(\hat{\theta})$ . On peut alors remplacer  $V_{MC}(\hat{\theta})$  en (B.6) par  $EQM_{MC}(\hat{\theta})$ .

(d) La probabilité de couverture d'un intervalle de confiance de niveau  $1 - \alpha$  % peut être approximée par

$$PC_{MC} = \frac{1}{R} \sum_{j=1}^R A_j \times 100 \quad (\text{B.8})$$

où  $A_j$  est une variable indicatrice telle que  $A_j = 1$  si l'intervalle de confiance pour le  $i^e$  échantillon, contient le paramètre  $\theta$  et  $A_j = 0$  sinon,  $j = 1, \dots, R$ .

### Choix du nombre d'itérations $R$

Une des aspects importants à considérer dans toute étude de simulation est le choix du nombre d'itérations  $R$ . Bien sûr, la qualité des approximations Monte Carlo augmente à mesure que le nombre d'itération  $R$  augmente. Cependant, pour de grandes valeurs de  $R$ , le temps d'exécution peut être excessivement long et/ou la mémoire de l'ordinateur peut s'avérer insuffisante. Il s'agit donc de déterminer une valeur appropriée de  $R$  qui permet d'obtenir une approximation de bonne qualité tout en respectant les contraintes informatiques.

Une méthode simple pour choisir le nombre d'itérations consiste à lancer la simulation plusieurs fois avec différentes valeurs de  $R$  et effectuer un examen des graphiques suivants :

- (i) un graphique du biais relatif Monte Carlo en fonction du nombre d'itérations  $R$ .
- (ii) un graphique de l'erreur quadratique moyenne Monte Carlo en fonction du nombre d'itérations  $R$ .

On choisira une valeur de  $R$  pour laquelle le biais relatif Monte Carlo et/ou l'erreur quadratique moyenne Monte Carlo est stable.

Il est important de noter que le nombre d'itérations nécessaire dépendra généralement de la taille de l'échantillon  $n$ . En effet, supposons que nous avons déterminé qu'il faille utiliser  $R = 1000$  itérations avec  $n = 100$  afin d'obtenir la précision désirée. Si nous décidons plutôt de tirer des échantillons de taille  $n = 50$ , le nombre d'itérations nécessaire, pour une précision donnée, sera vraisemblablement considérablement plus petit.

## ANNEXE C - Macro pour la répartition

Le code nécessaire pour répartir la taille échantillonnale entre les strates vous est fourni sous forme de **macro SAS**. Ainsi, il suffit de compiler le code une seule fois pour pouvoir l'utiliser à volonté et de façon générale. Une macro commence toujours par l'instruction **%MACRO** qui permet de nommer la macro et d'énumérer ses paramètres d'entrée. Elle se termine par **%MEND**. Une macro peut contenir des étapes DATA, des étapes PROC et des instructions macro précédées du symbole %. Les variables macro (comme celles prises en entrée) sont précédées du symbole &.

```
%MACRO repartition(pop=,type=,idstrate=,interet=,n=);
```

```
*pop : nom de la population;
```

```
*type : neyman ou prop;
```

```
*idstrate : variable indicatrice de strate;
```

```
*interet : variable d'intérêt;
```

```
*n : taille de l'échantillon;
```

```
/*On écrit les tailles  $N_h$  dans une table SAS*/
```

```
proc sort data=&pop;
```

```
    by &idstrate;
```

```
run;
```

```
proc means data=&pop n noprint;
```

```
    by &idstrate;
```

```
    output out=nobsPerStrata(drop=_FREQ_ _TYPE_) n=N_h;
```

```
run;
```

```
/*On calcule l'écart-type de la variable d'intérêt dans chaque strate de la population (utile pour la répartition de Neyman)*/
```

```
proc means data=&pop std noprint;
```

```

var &interet;
by &idstrate;
output out=StrataStd (drop =_TYPE_ _FREQ_) std=StrataStd;
run;

/*On procède à la répartition de la taille*/

data temp;
merge nobsPerStrata StrataStd;
by &idstrate;
%IF &type=prop %THEN %DO;
    StrataStd=1;
%END;
prod=N_h*StrataStd;
run;

proc sql noprint;
create table all as select *, &n*prod/sum(prod) as alloc, (calculated alloc)/N_h as
prop, 0 as takeall from temp;

/*On utilise une procédure itérative pour assurer que  $n_h$  ne dépasse pas  $N_h$  pour tout
 $h$ */
select max(prop) into :big_prop from all;
%DO %WHILE (&big_prop > 1);
    update all set takeall = 1 where prop > 1;
    create table all as select *, (&n - sum(takeall*N_h))*prod/sum((1-
takeall)*prod) as alloc, (calculated alloc)/N_h as prop from all (drop =
alloc prop);
    select max(prop) into :big_prop from all where takeall = 0;
%END;
quit;

/*On s'assure que la somme des tailles  $n_h$  donne  $n$ */

```

```

data all;
    set all (drop = prop);
    if takeall = 1 then alloc = N_h;
    roundalloc=round(alloc,1);
    diff=alloc-roundalloc;
run;

proc summary data=all;
    var alloc roundalloc;
    output out=checksum(drop=_TYPE_ _FREQ_) sum=orig round;
run;

data _NULL_;
    set checksum;
    diff=orig-round;
    call symput("DIFF",left(put(diff,5.0)));
run;

proc sql;
    create table allocation1 as
    select *
    from all, checksum;
quit;

%LET boolean=0;

%IF &diff GT 0 %THEN %DO;
    %LET boolean=1;
    proc sort data=allocation1;
        by takeall roundalloc;
    run;

    data allocation1;
        set allocation1;

```

```

        adjusted=roundalloc;
        if _N_ LE &DIFF then adjusted=adjusted+1;
    run;
%END;
%ELSE %IF &diff LT 0 %THEN %DO;
    %LET boolean=1;
    proc sort data=allocation1;
        by takeall descending roundalloc;
    run;

    data allocation1;
        set allocation1;
        adjusted=roundalloc;
        if _N_ LE abs(&DIFF) then adjusted=adjusted-1;
    run;

%END;

%IF &boolean=1 %THEN %DO;
    data allocation2;
        set allocation1;
        _NSIZE_=adjusted;
    run;
%END;
%ELSE %DO;
    data allocation2;
        set allocation1;
        _NSIZE_=roundalloc;
    run;
%END;

data allocation2;
    set allocation2;
    keep _NSIZE_ &idstrate;

```

`run;`

`%MEND; /*terminer la macro*/`

`/*Le fichier allocation2 contient les tailles échantillonnales finales*/`