SAS (3)

1. Import de données

1.1. Proc import

L'import dans SAS d'une table de format différent (.xls, .txt…) peut se faire soit par l'écriture d'un programme, avec une procédure **proc import**, soit par ouverture d'une boîte de dialogue spécifique (équivalant à un **proc import**).

Nous verrons cette seconde méthode, qui permet ensuite de récupérer le programme d'import correspondant.

→ Dans « file », cliquer sur « import data ».

→ Ouverture d'une boîte de dialogues avec étapes successives :

- Standard data source : sélectionner le format de la base de données source

- Where is the file located ? : sélectionner le fichier à importer. Le bouton "options" permet d'accéder à une 2ème boîte de dialogue :
  - « worsheet/range » : sélectionner la feuille où se trouve les données (fichier Excel notamment)
  - Case « column names in first row » : la cocher lorsque la première ligne du fichier importé contient les titres des colonnes.

- Choose the SAS direction : Library : sélectionner la librairie où sera importée la table

  Member : donner un nom SAS à la table importée.

- La dernière étape permet de créer un fichier programme sauvegardé sur l'ordinateur correspondant à la procédure d'import réalisée.
  - L'intérêt de cette étape est que ce fichier programme pourra ensuite être complété puis ouvert directement, sans avoir à repasser par les fenêtres d'import.

Une fois ouvert, ce programme s'affiche sous la forme suivante :

```
PROC IMPORT OUT= LIBRAIRIE.table
        DATAFILE= "chemin d'accès du fichier à importer"
        DBMS=FORMAT D'IMPORT REPLACE;
   GETNAMES=YES;
RUN;
```

Ex :

```
PROC IMPORT OUT= WORK.embol
        DATAFILE= "U:\_PUBLIC\sas\EmbolAMA7.xls"
        DBMS=EXCEL4 REPLACE;
   GETNAMES=YES;
RUN;
```

OUT = table créée dans SAS
DATAFILE = fichier à importer
DBMS = format du fichier à importer (NB : EXCEL4 = Excel 1997 ou 2000)
GETNAMES = YES : signifie qu'il faut conserver la première ligne, qui contient les noms des variables.

Ce programme peut également être écrit directement dans SAS.

1.2. Conseils pour mise en forme de fichier Excel avant import dans SAS

- Remplacer oui/non (variable alphanumérique) par 0/1 (variable numérique).
- Dates : doivent être au format dd/mm/yyyy
- Remplacer les virgules par des points en cas de problèmes d'import
- Remplacer les valeurs manquantes par des vides. SAS remplacera ensuite ces vides :
  - par un point (variable numérique)
  - par un blanc (variable alphanumérique)
- Si SAS ne reconnaît pas la dernière colonne d'une table Excel (parce qu'elle contient trop peu de valeurs par exemple) : rajouter une colonne fictive entièrement remplie à la fin de la table Excel.

Les bonnes pratiques de remplissage d'une table Excel sont également accessibles sur le site du laboratoire de santé publique de Tours, lien internet :

http://santepublique.med.univ-tours.fr/aidemethodo.html

→ Dans 2- Saisissez correctement vos données, cliquer sur « faire un fichier de données Excel »

1.3. Import dans l'étape data

L'import de tables de format différent peut également se faire dans l'étape data, pour les fichiers texte (.txt), plus pratiques à importer que Excel.

Ex :

```
data embolse;
infile "U:\_PUBLIC\sas\embol.txt" dlm='09'x missover dsd firstobs=10 obs=20
lrecl=310
format DN DEmbol DDC ddmmyy10.
informat DN DEmbol DDC ddmmyy10.
input n nompren $ DN  Dembol DDC;
if _n_>1 then output;
run;
```

Ici :

- On indique à SAS de créer une table temporaire nommée embolse.

Ligne infile :

```
infile "U:\_PUBLIC\sas\embol.txt" dlm='09'x missover dsd firstobs=10 obs=20
lrecl=310
```

- Infile : indique le chemin d'accès du fichier source à importer.
- Dlm= : indique le type de délimiteur entre chaque donnée.
  - ' ; ' : indique que le délimiteur est un point-virgule
  - '09'x : indique que le délimiteur est une tabulation (fichiers .xls, .csv, .txt)

- dsd : indique à SAS que lorsqu'il rencontre deux délimiteurs successifs, il remplace par une valeur manquante.

  NB : la valeur manquante sera un point (.) pour une variable numérique, un blanc pour une variable alphanumérique.

- Firstobs=10 : indique à SAS de garder les observations à partir de la 10ème.
- Obs = 20 : indique à SAS de garder 20 observations (20 premières par défaut, sinon à partir de celle indiquée en firstobs).

- Lrecl= : représente la longueur en nombre de caractères d'une case/ligne ?. lrecl = 256 par défaut, si l'on veut plus de caractères ceci doit être spécifié. En cas d'oubli et que le nombre de caractères est supérieur à 256, SAS passe à la ligne suivante et affiche une erreur dans le log. Il faut donc toujours penser à vérifier le log.

Ligne format :

```
format DN DEmbol DDC ddmmyy10.
```

Spécifie le format d'entrée des variables importées. Ici, le format des variables D N, Dembol et DDC est une date du type jourjourmoismoisannéeannée à 10 caractères.

Ligne informat :

```
informat DN DEmbol DDC ddmmyy10.
```

Spécifie le format de sortie dans SAS. Ici on n'a pas changé le format.

Ligne input

```
input n nompren $ DN  Dembol DDC;
```

Permet de préciser quelles sont les variables que SAS doit conserver. Les variables alphanumériques doivent être suivies d'un dollar ($).

NB : en l'absence de délimitation, il faut préciser le nombre de caractères de chaque variable, précédé d'un dollar s'il s'agit d'une variable alphanumérique.

Ex :

```
input finess $9. formatrsa $3. clersa $10.  nbrum 2.  ansortie 4.
```

Dernière ligne

```
if _n_>1 then output;
```

Ne se précise que lorsque la première ligne contient les noms de variables.

2. Proc tabulate

Permet de générer des tableaux de sortie de une à plusieurs dimensions par le croisement de plusieurs variables en lignes ou en colonnes.
Le procédure proc tabulate permet aussi de projeter plusieurs statistiques (effectifs, pourcentages, moyennes...).

2.1. Ecriture

proc tabulate data=table sur laquelle on travaille;
class var1 var2 var3…;
var var4 var5 var6… ;
table (cf remplissage ci-dessous);
run;

class : pour déclarer toutes les variables **qualitatives**
var : pour déclarer toutes les variables **quantitatives**
table : pour définir jusqu'à 3 dimensions de croisement des variables : colonne, ligne et page, qui est une tierce variable de découpage pour laquelle on sort autant de tableaux que de modalités de la variable.

Lorsque l'on définit les dimensions, l'ordre d'écriture des dimensions doit être le suivant : page, puis ligne, puis colonne. **Chaque dimension doit être séparée de la suivante par une virgule**.

Cependant, on n'est pas obligé de préciser à chaque fois 3 dimensions. Par défaut, si l'on écrit :

- une dimension seulement, il s'agira de la dimension colonne;
- deux dimensions, il s'agira des dimensions ligne puis colonne ;
- trois dimensions : page, puis ligne, puis colonne.

2.2. Exemple 1

```
proc tabulate data=greffe2;
class an sexe dom2;
var age duree_rss;
table sexe='Sexe'*age=''*(mean='Age moyen'*f=commax5.1 min='age
mini'*f=commax3.0 max='age maxi'*f=commax3.0) (dom2 all)*duree_rss=''*(n='Nb
séjours'*f=commax4.0  sum='Nb journées'*f=commax6.0
mean='DMS'*f=commax5.1), an='Année' all='Total';
run;
```

Ici :

- on travaille sur la table existante greffe2
- en utilisant les variables qualitatives an, sexe et dom2 (ligne class)
- et les variables quantitatives age et duree_rss (ligne var).

- table : Ici, il a été déterminé deux dimensions, séparées par une virgule :

  - Dimension 1 :

```
sexe='Sexe'*age=''*(mean='Age moyen'*f=commax5.1 min='age
mini'*f=commax3.0 max='age maxi'*f=commax3.0) (dom2 all)*duree_rss=''*(n='Nb
séjours'*f=commax4.0  sum='Nb journées'*f=commax6.0
mean='DMS'*f=commax5.1),
```

Ceci est la dimension ligne, qui va afficher :

- le sexe, stratifié par moyenne, minimum et maximum de l'âge.
- puis la variable dom2 et le total all (tous départements confondus), stratifiés par nombre de séjours, nombre de journées et DMS.

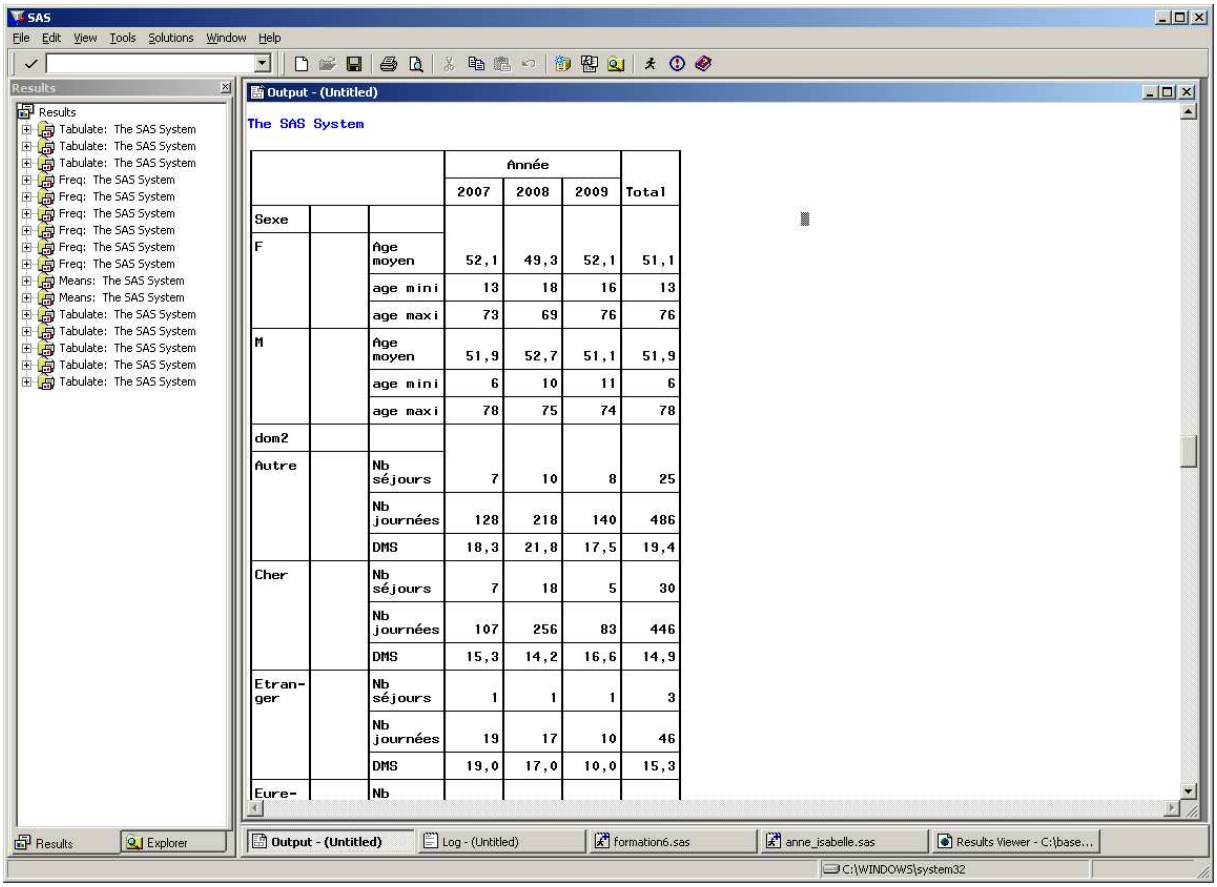
  - Dimension 2 :

```
an='Année' all='Total';
```

Ceci est la dimension colonne, où s'affichent :

- les années
- et le total, toutes années confondues

L'exécution du programme ci-dessus va ainsi donner la fenêtre de résultat suivante :



			Année			
			2007	2008	2009	Total
Sexe						
F		Age moyen	52,1	49,3	52,1	51,1
		age mini	13	18	16	13
		age maxi	73	69	76	76
M		Age moyen	51,9	52,7	51,1	51,9
		age mini	6	10	11	6
		age maxi	78	75	74	78
dom2						
Autre		Nb séjours	7	10	8	25
		Nb journées	128	218	140	486
		DMS	18,3	21,8	17,5	19,4
Cher		Nb séjours	7	18	5	30
		Nb journées	107	256	83	446
		DMS	15,3	14,2	16,6	14,9
Etran-ger		Nb séjours	1	1	1	3
		Nb journées	19	17	10	46
		DMS	19,0	17,0	10,0	15,3
Eure-		Nb				

On voit qu'ici SAS n'a sorti qu'un seul tableau (pas de dimension page définie), avec :

- en ligne :
  - Le sexe, stratifié par âge moyen, mini et maxi
  - puis dom2, stratifié en nombre de séjours, nombre de journées et DMS.

- en colonne, l'année et le total, toutes années confondues.

2.3. Remarques (exemple 1)

- L'astérisque (*) signifie que l'on croise les variables.
- L'utilisation de parenthèse suit la loi de distributivité.

Ainsi, la commande suivante :

```
sexe='Sexe'*age=''*(mean='Age moyen'*f=commax5.1 min='age
mini'*f=commax3.0 max='age maxi'*f=commax3.0),
```

est équivalente à :

```
sexe='Sexe'*age=''*mean='Age moyen'*f=commax5.1
sexe='Sexe'*age=''*min='age mini'*f=commax3.0
sexe='Sexe'*age=''*max='age maxi'*f=commax3.0
```

- Les intitulés des variables apparaissant dans le tableau de sortie sont indiqués entre quotes (' '). Ex : age=' ' : ceci indique que la variable age apparaîtra avec un intitulé vide.

- pour appliquer un format de sortie aux données, on utilise également l'astérisque *.
  Ex : *f=commax3.0 : on applique à la variable sélectionnée (intitulée « age mini ») le format commax3.0 : affichage de 3 caractères, dont aucun après la virgule.

- Ne pas confondre **variable** et **statistique** : une variable correspond à un intitulé de colonne dans la base de données (ex : age), une statistique correspond à une fonction appliquée à une variable (ex : mean, min, max)

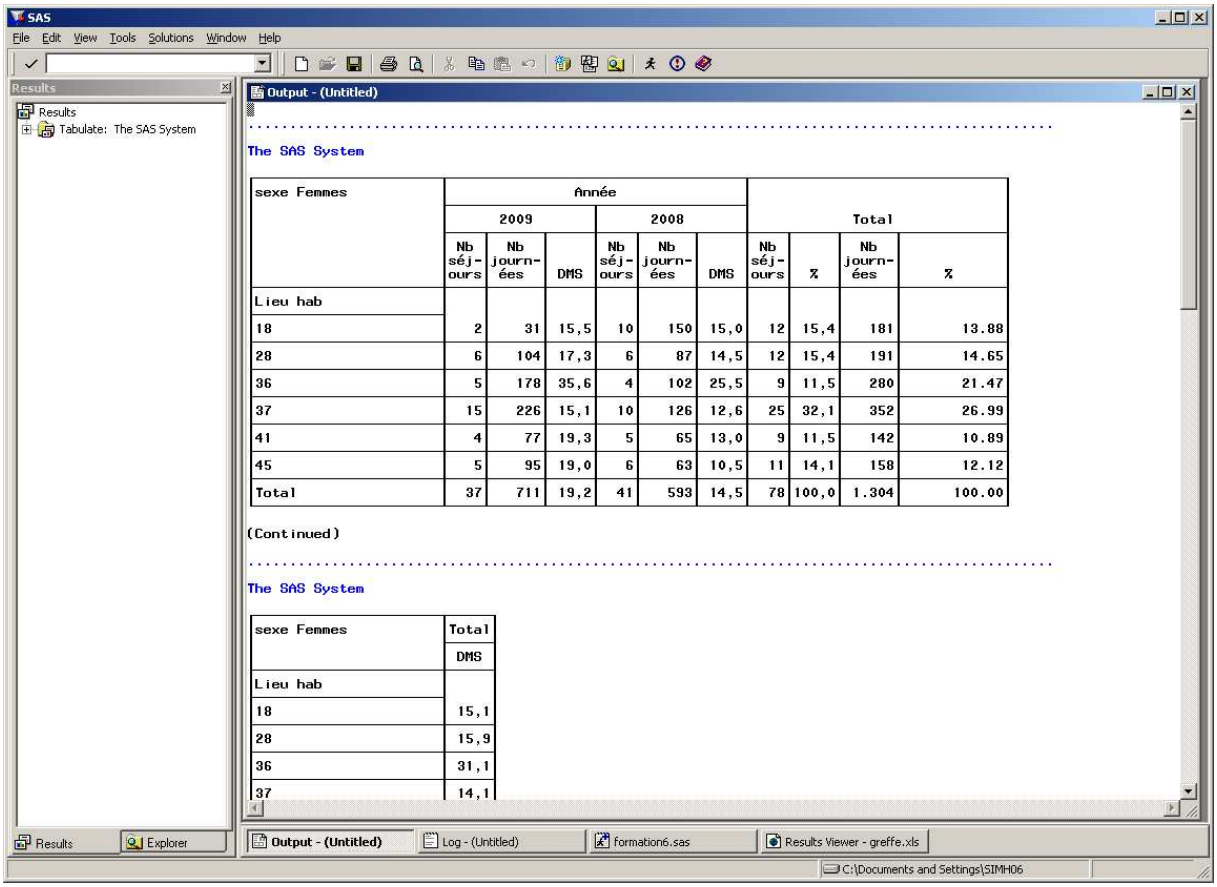
2.4. Exemple 2

```
proc tabulate data=greffe2 missing;
where an in (2008,2009) and dom2 in('Cher','Eure-et-Loire','Indre','Indre-et-
Loire','Loir-et-Cher','Loiret');
class an/descending ; class dom sexe;
var duree_rss;
table sexe,
dom all,
an*duree_rss=''*(n*f=commax4.0   sum*f=commax6.0  mean*f=commax5.1)
all*duree_rss=''*(n*f=commax4.0  colpctn='%'*f=commax5.1 sum*f=commax6.0
colpctsum='%' mean*f=commax5.1)/box=_page_;
label an='Année' dom='Lieu hab';
keylabel all='Total' n='Nb séjours' sum='Nb journées' mean='DMS';
format sexe $sexe. ;
run;
```

- Ici, on travaille à partir de la table greffe 2, pour laquelle on va conserver les données manquantes (missing)
- On sélectionne les valeurs numériques 2008 et 2009 de la variable an et les départements qui nous intéressent de la variable dom2.
- Ligne class : on définit les variables qualitatives an, dom et sexe. An sera classée par ordre décroissant (/descending).

- Ligne var : on définit la variable quantitative duree_rss
- Ligne table :
    - En page, apparaîtra la variable sexe
    - en ligne, apparaîtront la variable domicile et le total, tous départements confondus
    - En colonne, apparaîtront par année et toutes années confondues (Total), la durée de rss, stratifiée par nombre de séjours (n), nombre de journées (sum), et DMS (mean)

Le résultat apparaît sous la forme suivante :



SAS a ici sorti plusieurs tableaux (=pages, cf dimension page), chaque tableau correspondant à une modalité de la variable sexe : hommes, femmes , total.

2.5. Remarques (exemple 2)

- Pour conserver les données manquantes : indiquer missing dans la ligne du **proc tabulate.**

- Pour trier les sorties selon un ordre décroissant : /descending ou croissant : /ascending ou encore order=freq ... sur ce qu'on souhaite. C'est une option dans la ligne class.

On peut écrire plusieurs lignes class, séparées par des points-virgules.

Dans la ligne table :

- Colpct + nom de la variable ou statistique : ajoute le pourcentage en colonne pour la variable ou statistique souhaitée.
  Ex : colpctn : ajoute le pourcentage en colonne pour la statistique n.
  colpctsum : idem pour la statistique sum

  Il existe la même chose pour les pourcentages en ligne : rowpct…

- /box=_page_ : fait apparaître le libellé de la dimension page correspondante.
  Si l'on souhaite faire apparaître un libellé spécifique, il faut écrire :
  / box = 'texte'

- On peut appliquer les libellés différemment que dans l'exemple précédent :

Ex :

```
label an='Année' dom='Lieu hab';
```

→ spécifique aux variables

```
keylabel all='Total' n='Nb séjours' sum='Nb journées' mean='DMS';
```

→ spécifique aux statistiques

- On peut également appliquer différemment un format :

Ex :

```
format sexe $sexe.
```

Le format doit être défini en-dehors du **proc tabulate** :
Ex :

```
proc format; value $sexe F='Femmes' M='Homme';run ;
```

3. Export des données (rappel)

Export en Excel :

```
ods html file="C:\bases\greffe.xls";
programme;
ods html close;
```

NB :

Html = format excel
Rtf = format word